



Control Visual de un Brazo Manipulador con 7GDL, en Base a Visión Monocular, para el Seguimiento de Objetivos

Maria Fernanda Tejada Begazo

Orientador: Prof Dr. Dennis Barrios Aranibar

Tesis profesional presentada a la Escuela Profesional de Ciencia de la Computación como parte de los requisitos para obtener el Título Profesional de Ingeniera en Informática.

UCSP- Universidad Católica San Pablo
Abril de 2018

A Dios, por darme la fuerza para continuar con esta investigación. A mi familia, por comprender el tiempo que le he dedicado a este trabajo. Y a mis compañeros y profesores del grupo de investigación por su compañía y enseñanzas.

Abreviaturas

CNN *Convolutional Neural Networks*

D-H Denavit-Hartenberg

FEM *Finite Element Method*

FINCyT Fondo para la Innovación, Ciencia y Tecnología

FPGA *Field Programmable Gate Array*

FPR Falsos Positivos

GDL Grados de Libertad

H-R Hayati-Roberts

HSL *Hue, Saturation and Lightness*

IBVS *Visual Servoing* basado en Imágenes

IFR Federación Internacional de Robótica

KF Filtro de Kalman

LARVIC Grupo de Investigación en la Línea de Automatización Industrial, Robótica y Visión Computacional

PBVS *Visual Servoing* basado en Posiciones

PLC *Programmable Logic Controller*

ROC *Receiver Operating Characteristic*

ROS *Robot Operating System*

UCSP Universidad Católica San Pablo

VPR Verdaderos Positivos

WOFC *Wise Optical Flow Clustering*

Agradecimientos

A Dios, por darme la fuerza y habilidades para realizar esta investigación.

A mis padres y hermano, por comprenderme en los momentos que no pasaba tiempo con ellos, los consejos y las fuerzas que me han dado durante la elaboración de esta tesis.

Al Fondo para la Innovación, Ciencia y Tecnología (FINCyT), por el soporte económico brindado al proyecto (contrato 2016-FINCyT-IA-2013).

A mi asesor, Dr. Dennis Barrios y mi co-asesora, Dra. Raquel Patiño, por su apoyo, palabras y dedicación a la realización de este trabajo de investigación.

A Claudia Cervantes, por sus ánimos y enseñanzas en el transcurso de este trabajo de investigación.

A mis revisores, por sus correcciones que me ayudaron a mejorar este proyecto de investigación.

A la Universidad Católica San Pablo (UCSP), por brindar las instalaciones y apoyo institucional, para lograr las diferentes tareas relacionadas al desarrollo del presente trabajo.

A mis amigos y compañeros del Grupo de Investigación en la Línea de Automatización Industrial, Robótica y Visión Computacional (LARVIC) de la UCSP, por permitirme tener el honor de compartir proyectos, desveladas, presiones y angustias; pero también momentos de gran satisfacción y buenos paseos, mis respetos a todos y cada uno de ellos, les agradezco a todos los miembros el haberme permitido hacer equipo a su lado.

A Elizabeth Morales Muñoz, mi amiga, que a través de sus palabras de aliento me ha impulsado a seguir adelante.

Por último, a Kevin Rodríguez, Jhilver Quispe y Liz Bernedo por comprender la poca disponibilidad de tiempo para los proyectos que nos hemos propuesto en estos años (Larvic-Saurus y Larvic-Prime).

Resumen

La necesidad de incrementar la producción de las grandes empresas en la Primera Revolución Industrial permitió el desarrollo de nuevas máquinas, tecnologías y actividades, configurando el entorno perfecto para la aplicación de máquinas y procedimientos autónomos como los brazos manipuladores. En la última década se han ampliado las actividades que realizan los brazos manipuladores a diversas áreas como rescate, medicina e industria aeroespacial. La principal tarea de un brazo manipulador es alcanzar un objetivo por medio de sus elementos perceptivos. Esta tarea conlleva escoger los sensores necesarios para percibir el mundo tomando en cuenta el costo, el peso y el espacio. En esta investigación se dará solución a este problema con el uso de un sensor de visión, es decir una cámara. El mecanismo de control que se presenta se basa en dividir el movimiento tridimensional en dos movimientos sobre dos planos: Uno de estos planos es el mismo que el plano de la cámara (plano XY) y el otro plano será perpendicular al primero y se refiere a la profundidad (plano XZ). El movimiento del objetivo en el plano de la cámara será calculado por medio del flujo óptico, es decir la traslación del objetivo del tiempo t al $t + 1$ en el plano XY . En cambio, el movimiento en el plano de la profundidad se estimará mediante el filtro de Kalman usando las variaciones de la traslación obtenida del flujo óptico y de la rotación dada por la matriz de cinemática directa. Finalmente, el movimiento planificado en cada plano se ejecutará de forma intercalada infinitesimalmente, obteniendo así un movimiento continuo para los tres ejes coordenados (XYZ). Los resultados experimentales obtenidos, han demostrado que se realiza un camino limpio y suavizado. Se han llevado a cabo pruebas con diferentes intensidades de iluminación, mostrando un error promedio de la trayectoria de movimiento de $\mu_{x,y,z} = 5.05, 4.80, 3.0$ en centímetros con iluminación constante, por lo que se tiene una desviación estándar $\sigma_{x,y,z} = 2.21, 2.77, 1.45$ en centímetros. Al obtener resultados satisfactorios en las pruebas elaboradas. Se puede concluir que es posible solucionar el problema del movimiento tridimensional de un brazo manipulador dividiéndolo en dos sub-problemas que trabajan en planos perpendiculares. Esta solución nos proporciona una trayectoria suave, ya que el mecanismo de control se realiza en cada instante de tiempo obteniendo un movimiento natural.

Abstract

The need to increase the production of large companies in the First Industrial Revolution allowed the development of new machines, technologies and activities, configuring the perfect environment for the application of machines and autonomous procedures such as manipulator arms. In the last decade the activities carried out by the manipulator arms have been expanded to various areas such as rescue, medicine and the aerospace industry. The main task of a manipulator arm is to reach a goal by means of its perceptual elements. The task is to choose the sensors necessary to perceive the world taking into account the cost, weight and space. This research will solve this problem with the use of a vision sensor, a camera. The control mechanism to be presented is based on dividing the three-dimensional motion into two movements on planes: One of these planes is the same as the plane of the camera (plane XY) and the other plane will be perpendicular to the first plane and refers to the depth (plane XZ). The movement of the lens in the plane of the camera will be calculated by means of the optical flow, that is, the translation of the time objective t to $t + 1$ in the XY plane. In contrast, the motion in the plane of the depth will be estimated by the Kalman filter using the variations of the translation obtained from the optical flow and the rotation given by the direct kinematic matrix. Finally, the movement planned in each plane will be executed infinitesimally in an interleaved way, thus obtaining a continuous movement for the three coordinate axes (XYZ). The experimental results obtained have shown that a clean and smoothed path is carried out. Tests have been carried out with different illumination intensities, showing an average error of the motion path of $\mu_{x,y,z} = 5.05, 4.80, 3.0$ in centimeters with constant illumination, so we have a standard deviation $\sigma_{x,y,z} = 2.21, 2.77, 1.45$ in centimeters. By obtaining satisfactory results in the elaborated tests. It can be concluded that it is possible to solve the problem of three-dimensional movement of a manipulator arm by dividing it into two sub-problems working on perpendicular planes. This solution provides a smooth path, since the control mechanism is performed at every instant of time obtaining a natural movement.

Contenido

1	Introducción	2
1.1	Motivación y Contexto	2
1.2	Planteamiento del Problema	3
1.3	Objetivos	4
1.3.1	Objetivos Específicos	4
1.4	Organización de la tesis	4
2	Fundamentos del Sistema de Control Visual para Brazos Manipuladores	6
2.1	Estructura y Clasificación de Brazos Manipuladores	6
2.2	Visión Computacional	8
2.2.1	Procesamiento de Imágenes	8
2.2.2	Extracción de características	11
2.3	Flujo óptico	13
2.4	Estimación de Profundidad	14
2.5	<i>Visual Servoing</i>	17
2.5.1	<i>Visual-Motor Model Estimation</i>	19
2.5.2	Estimación de Control [Kragic and Christensen, 2002]	20
2.6	Proceso de Control de Brazos Manipuladores	21
2.6.1	Parámetros de un Brazo Manipulador	23
2.6.2	Cinemática Directa	24
2.6.3	Lectura de Sensores	27

2.6.4	Estrategia para la siguiente posición	27
2.6.5	Calcular camino libre de obstáculos y puntos de control	27
2.6.6	Cinemática Inversa	28
2.7	Consideraciones Finales	33
3	Estado del Arte	35
3.1	<i>Visual Servoing</i>	35
3.2	Técnicas de Modelado de Brazos Manipuladores	36
3.3	Flujo Óptico	39
3.4	Estimación de Profundidad	41
3.5	Cinemática Inversa	42
4	Propuesta	43
4.1	Estrategia "Divide y Vencerás"	43
4.1.1	Calcular la Posición XY del Objetivo	44
4.1.2	Estimar la Profundidad	46
4.1.3	Movimiento de la Articulación	48
5	Resultados	49
5.1	Parámetros de Implementación	49
5.1.1	Hardware	49
5.1.2	Software	50
5.1.3	Variables de implementación	51
5.2	Pruebas Globales	53
5.3	Pruebas Específicas	57
5.3.1	Pruebas de Percepción	57
5.3.2	Cinemática Directa	60
6	Conclusiones y Trabajos Futuros	64

6.1	Conclusiones	64
6.2	Problemas encontrados	65
6.3	Recomendaciones	65
6.4	Limitaciones	66
6.5	Trabajos futuros	66
Bibliografía		73

Lista de Tablas

2.1	Parámetros de D-H para el brazo manipulador cilíndrico (Fig. 2.12).	26
2.2	Parámetros de D-H del brazo manipulador de la Fig. 2.14.	30
5.1	Parámetros de D-H de la base a la cámara del brazo manipulador del Komodo.	53
5.2	Promedio y Desviación estándar de los casos de estudio en cm.	54
5.3	Diferentes perspectivas de iluminación de la esfera.	59
5.4	Casos de pruebas de la etapa de percepción en diferentes fondos.	60
5.5	Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en <i>base_rotation</i> .	61
5.6	Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en <i>shoulder</i> .	62
5.7	Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en <i>elbow1</i> .	62
5.8	Error promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en <i>elbow2</i> .	62
5.9	Error promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en <i>wrist</i> .	63

Lista de Imágenes

2.1	Morfología de un robot industrial [Ramirez, 2011].	7
2.2	Ejemplo de un brazo manipulador de 3 GDL (inspirado en [Alsina, 2014]). . .	8
2.3	Tipos de articulaciones para brazos manipuladores [Barrientos et al., 1997a]. . .	9
2.4	Esquema de procesamiento de la operación espacial en la imagen [Corke, 2017]	10
2.5	Reproducción del flujo óptico en la retina y las imágenes [Raudies, 2013]. . . .	13
2.6	Taxonomía de las restricciones para la estimación del flujo óptico [Raudies, 2013].	15
2.7	Vertientes fundamentales de los sistema de control de brazos manipuladores [Kragic and Christensen, 2002].	17
2.8	Estructura del <i>visual servoing</i> basado en posiciones (PBVS) [Corke, 1994]. . .	19
2.9	Estructura del control visual basado en imágenes (IBVS) [Corke, 1994].	20
2.10	Diagrama de bloques para el control de un brazo manipulador.	22
2.11	Representación de la notación Denavit-Hartenberg (D-H) [Spong et al., 2004a].	23
2.12	Brazo manipulador cilíndrico.	26
2.13	Geometría plana asociada al brazo manipulador de 3 Grados de Libertad (GDL) [Craig, 2006].	29
2.14	Sistema de referencias del brazo manipulador de 3 GDL [Barrientos et al., 1997b].	31
3.1	Representación del eje <i>screw</i> de Movimiento [Ajwad et al., 2014].	37
3.2	Convención de Hayati-Roberts [Ajwad et al., 2014].	38
4.1	Diagrama de la implementación del sistema de control visual para el brazo ma- nipulador del robot Komodo para alcanzar un objetivo.	44
4.2	Partes del primer componente de percepción del objetivo.	45

4.3	Partes de la Estimación de Profundidad (inspirado por [Zhuang et al., 1994]).	46
5.1	Robot Komodo [RoboTiCan, 2013].	50
5.2	Modulo del Brazo del Robot Komodo (fotografía del brazo manipulador del Robot Komodo).	50
5.3	Notación de D-H basado en la cámara del brazo manipulador del robot Komodo.	52
5.4	Casos de prueba según la iluminación y la posición.	54
5.5	Trayectoria del eje X del caso 1.	55
5.6	Trayectoria del eje Y del caso 1.	55
5.7	Trayectoria del eje Z del caso 1.	56
5.8	Trayectoria del eje X del caso 2.	56
5.9	Trayectoria del eje Y del caso 2.	57
5.10	Trayectoria del eje Z del caso 2.	57
5.11	Trayectoria del eje X del caso 3.	58
5.12	Trayectoria del eje Y del caso 3.	58
5.13	Trayectoria del eje Z del caso 3.	61
5.14	Curva de <i>Receiver Operating Characteristic</i> (ROC) de la percepción bidimensional del objeto en los diferentes casos de prueba.	61

Capítulo 1

Introducción

1.1 Motivación y Contexto

En la Primera Revolución Industrial surgió la necesidad de incrementar la producción de las grandes empresas; permitiendo el desarrollo de nuevas máquinas, tecnologías y actividades. Por otra parte, la naturaleza de estas nuevas tareas repetitivas y simples, perjudicaban la salud del ser humano. Por lo tanto, configuró el entorno perfecto para la introducción de la automatización industrial, es decir, la aplicación de máquinas y procedimientos autónomos para realizar funciones monótonas.

Las principales máquinas utilizadas en la automatización industrial fueron los brazos manipuladores, también llamados brazos robóticos. Generalmente, estas máquinas eran empleadas en la industria automovilística donde se realizaban tareas deterministas en un espacio de trabajo definido [Barrientos et al., 1997a]. De hecho, en el Laboratorio Nacional de Argonne de Estados Unidos, estas máquinas se aplicaban en la manipulación de elementos radioactivos desde 1948 [Paul, 1984].

Inicialmente los brazos manipuladores tenían un mecanismo de control simple, llamado maestro-esclavo. Este mecanismo de control consta de un brazo manipulador en escala que es controlado por una persona en un espacio seguro. Así, la máquina real sólo replica los movimientos en el espacio de trabajo [Goertz, 1964].

De este modo la empresa Hydro-Lek presenta un brazo manipulador de 5 GDL (HLK-HD5) usado por la marina italiana para la investigación y rescate marítimo. Este brazo robótico realiza inspección visual, recuperación de objetos y rescate bajo el agua [House and Road, 2014]. También, en las tareas de rescate se emplean brazos manipuladores autónomos [Alsina, 2014]. Así en el 2011, Bagnell y sus colegas realizaron un brazo manipulador de 7 GDL equipado con una mano de tres dedos y una percepción de un sensor láser y visión estéreo. Este brazo manipulador tiene un sistema capaz de analizar, identificar y realizar una trayectoria para alcanzar un objetivo [Bagnell et al., 2012], [Kazemi et al., 2012].

Los brazos manipuladores han evolucionado al pasar los años, obteniendo una mayor autonomía y precisión. Por consiguiente, se han desarrollado nuevas aplicaciones en diferentes áreas como la industria alimentaria [Mehta, 2007], en rescate [Kemp et al., 2007], la industria

aeroespacial [Belfiore and Benedetto, 2016] y la medicina [Osa et al., 2010, Azizian et al., 2014, Azizian et al., 2015].

Los sistemas de control de brazos manipuladores autónomos tienen dos vertientes. La primera vertiente es un mecanismo de control que se basa en percibir el objetivo y realizar el movimiento correspondiente, llamado lazo abierto. Este mecanismo después de percibir el objetivo, no vuelve a activar sus sensores perceptivos. Por lo que, se crea una nueva vertiente llamada lazo cerrado, es decir, un mecanismo de control que consta de la percepción del ambiente y del objetivo a cada instante de tiempo [Paul, 1984]. Por ejemplo, cuando un brazo robótico industrial recoge un objeto de una banda transportadora para trasladarlo a una caja; eventualmente, después ya iniciado el movimiento del brazo, la posición de la caja podría cambiar, por lo que el brazo manipulador debe percibir esta situación y modificar la trayectoria para completar su tarea.

A través del tiempo, se han dado diferentes propuestas para el control de lazo cerrado. Por ejemplo, en la industria aeroespacial se elaboró un brazo robótico para capturar objetos que están flotando [Yang et al., 2014]. Este brazo manipulador resuelve el problema de capturar objetos con un sistema de control visual en tiempo retardado. Además, existen brazos manipuladores móviles que tienen un mecanismo de control visual para llegar de una posición arbitraria a una ubicación objetiva [Nadi et al., 2014].

El sistema de control para un brazo robótico depende de muchos factores como: la estructura, la posición de los sensores, la cantidad de sensores, los tipos de sensores y la vertiente que se usará. Por ejemplo, un sistema de control para un brazo de 4 GDL es diferente al de 7 GDL, ya que la cantidad de soluciones posibles para el movimiento de una posición a otra difiere del número y del tipo de articulaciones. Además, si el sensor es una cámara ubicada en la base se limita el campo de visión, sin embargo, al estar posicionada en la garra se tiene mayor amplitud del campo de visión. Por último, las vertientes de lazo abierto se implementan de forma rápida, pero limitan la dinámica del mundo real, en cambio, un sistema de lazo cerrado considera la retroalimentación visual a cada instante de tiempo que permite modelar un mundo dinámico, es decir, más real.

A parte de observar la estructura y el posicionamiento de los sensores en el brazo manipulador, se debe escoger el tipo de sensor que se usará. Por lo que se debe tener dos características fundamentales: el rango de error y el propósito que va a cumplir el sensor. En este sentido, una cámara depende de su calibración para obtener el margen de error y nos permite seleccionar características de un objeto, seguir objetivos, posicionar el objetivo en el plano, etc.

Al examinar las anteriores pautas, se ha decidido realizar esta investigación en el control de un brazo manipulador con el uso de un sensor visual, es decir una cámara, para el seguimiento de un objetivo. El sensor visual estará posicionado en la garra del brazo manipulador, por lo que se implementará un sistema de control de lazo cerrado.

1.2 Planteamiento del Problema

Un brazo manipulador está limitado en peso, espacio y costo. Lo que tiene como consecuencia la existencia de diferentes estructuras para los brazos manipuladores y tipos de sensores.

Por este motivo, el principal problema de un brazo manipulador es implementar un sistema de control adecuado, teniendo en cuenta cada uno de los dos componentes mencionados con anterioridad.

Esta investigación considera solucionar el problema mencionado anteriormente para el caso específico de un brazo manipulador de 7 GDL, que posee una cámara en la garra como elemento perceptivo. Lo anterior introduce un nuevo problema que se plasma en la pregunta: ¿Cómo se puede implementar un sistema de percepción-control en un robot manipulador que debe actuar en un espacio de trabajo tridimensional y que percibe el mundo solo de forma bidimensional?

1.3 Objetivos

Implementar una estrategia que permita resolver el problema de poseer una percepción bidimensional del mundo tridimensional; de tal forma que posibilite controlar un brazo manipulador de 7GDL provisto de una cámara, anclada en su garra, para el seguimiento de un objetivo dentro de su espacio de trabajo.

1.3.1 Objetivos Específicos

Los objetivos específicos a alcanzar son:

- Elaborar algoritmos que permitan calcular la posición de objetivos en un espacio tridimensional a partir de información bidimensional.
- Desarrollar algoritmos de control de un brazo manipulador de 7GDL que permitan ejecutar movimientos independientes en dos planos perpendiculares (el plano de la cámara y el plano de la profundidad).
- Desarrollar una estrategia que permita fusionar los movimientos de control de un brazo manipulador de 7GDL, desarrollados en dos planos perpendiculares entre sí, de forma que se ejecute eficazmente un movimiento en su espacio de trabajo.

1.4 Organización de la tesis

La tesis está organizada con los siguientes capítulos:

Segundo capítulo, este capítulo se dividirá en cuatro partes. La primera parte detallará la estructura y clasificación de brazos manipuladores. Seguidamente, se analizará los diferentes sistemas de visión. La tercera parte explicará el proceso básico para realizar el sistema de control de un brazo manipulador. Por último, se dará las condiciones finales que se deben tener en cuenta para trabajar con un sistema de control basado en visión de un brazo manipulador.

Tercer capítulo, en este capítulo se examinará el estado del arte en el cual se basa la tesis. Se analizará las técnicas del modelado de brazos manipuladores, flujo óptico, estimación de la profundidad y cinemática inversa.

Cuarto capítulo, en este capítulo contempla la propuesta; en el cual se basa en la estrategia de “divide y vencerás” para solucionar el problema presentado. Luego, se detallará los puntos que se han implementado para realizar la propuesta.

Quinto capítulo, en este capítulo se observarán los parámetros de implementación y las pruebas realizadas.

Sexto capítulo, se analizarán los resultados de las pruebas realizadas. Luego, se detallará los problemas encontrados en la implementación y las limitaciones de la propuesta. Finalmente, se explicará los trabajos futuros que se pueden realizar en esta área.

Capítulo 2

Fundamentos del Sistema de Control Visual para Brazos Manipuladores

Este capítulo se dividirá en cuatro partes. En la primera parte se explicará las propiedades de brazos manipuladores. En la segunda parte, se analizará la teoría de los sistemas de control visual y cuáles son las técnicas que se están desarrollando. Se dará énfasis a la técnica de flujo óptico, utilizada para obtener el movimiento de un objetivo en la perspectiva de la cámara. Por último, se examinará las técnicas para la estimación de la profundidad.

En la tercera parte, se explica el proceso básico para realizar el sistema de control de un brazo manipulador. Por lo que, primero se examinará los parámetros para realizar el sistema de control. A parte de ello, se explicará las técnicas de cinemática directa y cinemática inversa. Por último, las consideraciones del cálculo de la trayectoria y los puntos de control que se emplean para obtener un camino libre de obstáculos.

Finalmente, se observa las condiciones que se deben tomar para realizar el sistema de control visual para un brazo manipulador. Por lo que, se da un punto de vista técnico para obtener un sistema de control adecuado.

2.1 Estructura y Clasificación de Brazos Manipuladores

Un brazo manipulador consta de articulaciones o accionadores que permiten el movimiento, que son unidos por medio de eslabones. Además, los brazos manipuladores en el extremo tienen una herramienta con la que realiza las tareas respectivas. La Figura 2.1 se puede observar la estructura simple de un brazo manipulador y la Figura 2.2 se percibe un esquema de los componentes principales.

Los accionadores son los motores de las articulaciones, de manera que cada articulación puede tener diferentes movimientos. Por ejemplo, en la Figura 2.3 se puede observar las diferentes clasificaciones de articulaciones. Sin embargo, en los brazos manipuladores modernos se emplean comúnmente los accionadores rotacionales permitiendo movimientos entre 0° a 180° . En cambio, los accionadores prismáticos proporcionan desplazamientos lineales

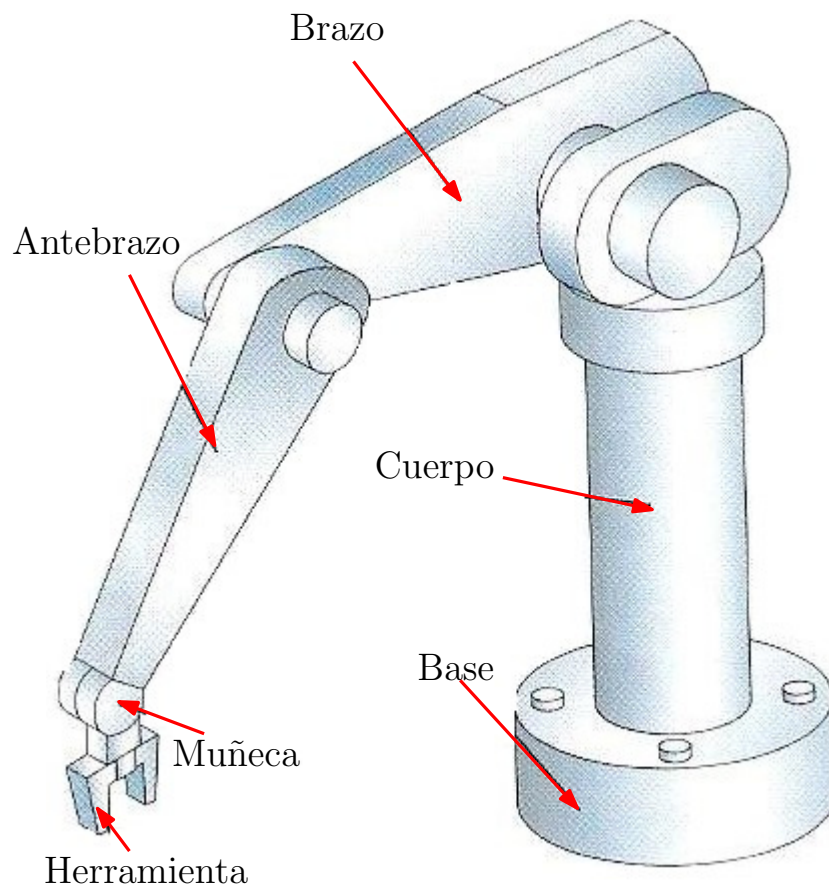


Figura 2.1: Morfología de un robot industrial [Ramirez, 2011].

[Spong et al., 2004b].

Para que una máquina sea considerada un brazo manipulador debe poseer las siguientes características [Alsina, 2014]:

- Presenta un punto fijo adherido a una base.
- Trata de simular los movimientos del brazo humano.
- Presenta un espacio de trabajo limitado.

En los brazos manipuladores se tiene el término Grados de Libertad (GDL por sus siglas en español), que representa la suma de los movimientos que puede realizar cada articulación. Por ejemplo, la Figura 2.2 representa un brazo robótico de 3 GDL para el posicionamiento. En cambio, un brazo de 6 GDL tiene tres articulaciones para realizar posicionamiento en los tres ejes coordenados y tres articulaciones para la orientación [Spong et al., 2004b]. En otros términos, GDL son los movimientos de un cuerpo rígido en el espacio tridimensional, es decir, los cambios de posición hacia delante-atrás, arriba-abajo e izquierda-derecha en los tres ejes perpendiculares. De igual importancia se tiene el cambio de la orientación a través de la rotación alrededor de los tres ejes: *Yaw*, *pitch* y *roll* (dirección, elevación y ángulo de alabeo).

En el informe técnico ISO/TR 83737 realizado por la Federación Internacional de Robótica (IFR) nos proporciona formalmente la clasificación de los brazos manipuladores en tipos como:

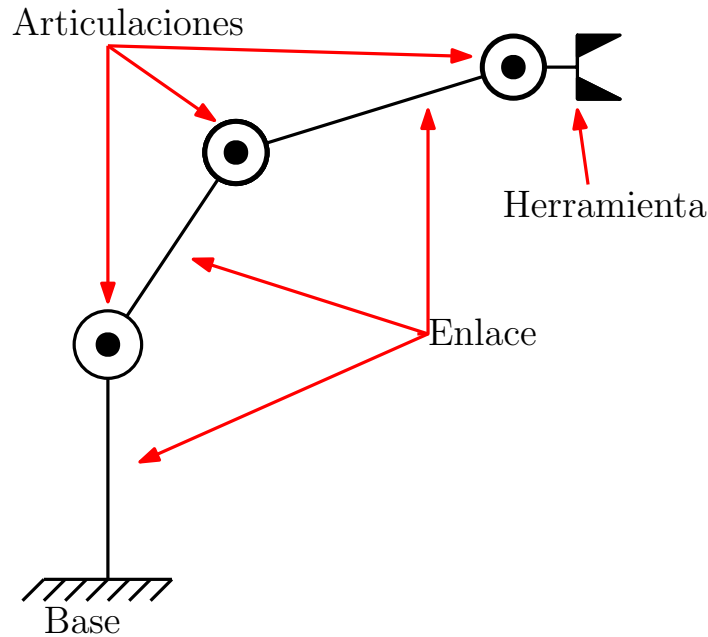


Figura 2.2: Ejemplo de un brazo manipulador de 3 GDL (inspirado en [Alsina, 2014]).

- Tipo A: Manipulador con control manual o tele-operado.
- Tipo B: Manipulador automático con ciclos pre-ajustados de control con *Programmable Logic Controller* (PLC) donde se puede realizar accionamiento neumático, eléctrico o hidráulico.
- Tipo C: Robots programables con trayectoria continua (punto a punto).
- Tipo D: Robots capaces de adquirir datos de su entorno para adaptar sus tareas en función de éstos datos.

2.2 Visión Computacional

La mayoría de autores persisten en que los sensores más efectivos son los ojos para reconocimiento, navegación, evitar obstáculos y manipulación. Las cámaras imitan esta función y por lo que se crea el control basado en visión para los brazos manipuladores, de este modo se usa imágenes digitales para reconocer objetos y navegar dentro de un ambiente [Corke, 2017].

2.2.1 Procesamiento de Imágenes

El procesamiento de imágenes es un proceso computacional que transforma una o más imágenes de entrada en una imagen de salida. Este procesamiento se utiliza con frecuencia para mejorar una imagen para la visualización o interpretación humana [Corke, 2017].

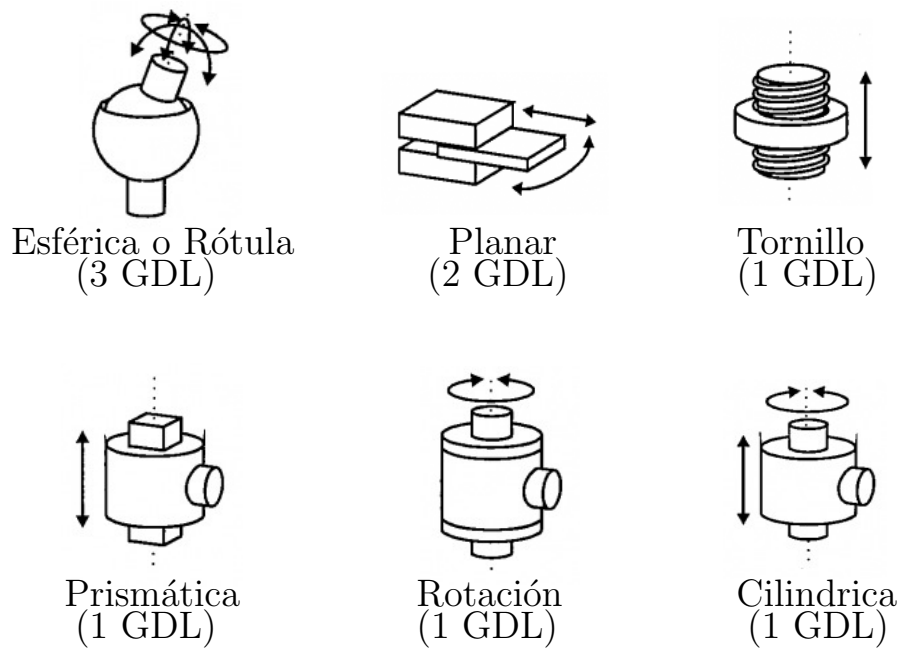


Figura 2.3: Tipos de articulaciones para brazos manipuladores [Barrientos et al., 1997a].

Operaciones Diagonales

Las operaciones diagonales, permiten que dos imágenes de entrada den como resultado una única imagen de salida, y las tres imágenes son del mismo tamaño. Cada píxel de salida es una función de los píxeles correspondientes en las dos imágenes de entrada.

$$O[u, v] = f(I_1[u, v], I_2[u, v]), \forall (u, v) \in I_1 \quad (2.1)$$

Distinguir los objetos en primer plano del fondo es un problema importante en la visión en robots, pero rara vez se tiene el lujo de tener un fondo especial. Por lo que, se toma una fotografía de la escena sin un objeto en primer plano presente y considera esto como el fondo, pero eso requiere tener un conocimiento especial sobre cuándo el objeto no está en primer plano. También, se asume que el fondo no varía con el tiempo, la variación es un problema en ambientes del mundo real donde la iluminación ambiental y las sombras cambian en intervalos de tiempo bastante cortos, y el ambiente puede modificarse estructuralmente a intervalos de tiempo muy largos.

Operaciones Espaciales

Las operaciones espaciales se muestran esquemáticamente en la Fig. 2.4, cada píxel en la imagen de salida es una función de todos los píxeles en una región que rodea el píxel correspondiente en la imagen de entrada [Corke, 2017], como se muestra en la ecuación 2.2.

$$O[u, v] = f(I[u + i, v + j]), \forall (i, j) \in W, \forall (u, v) \in I, \quad (2.2)$$

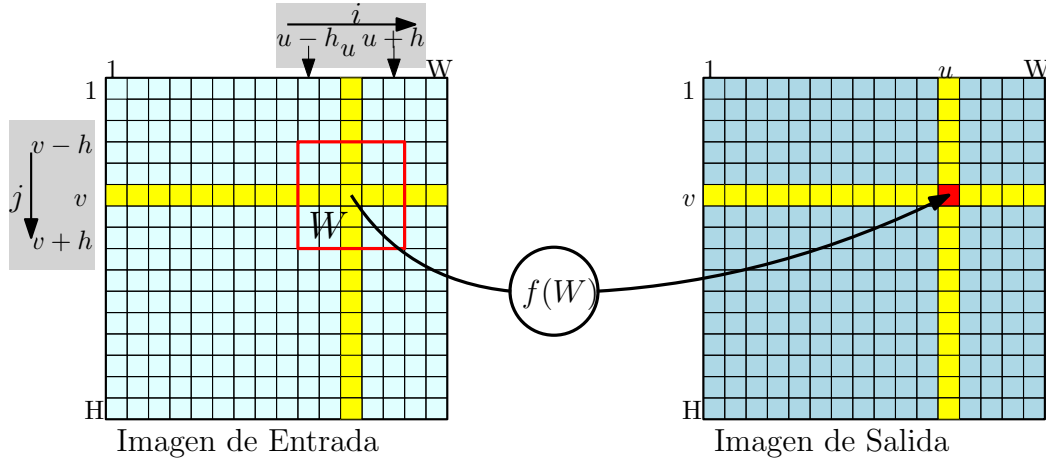


Figura 2.4: Esquema de procesamiento de la operación espacial en la imagen [Corke, 2017]

dónde W es conocido como ventana, típicamente una región cuadrada $w \times w$. Las operaciones espaciales son potentes debido a la variedad de posibles funciones f , lineales o no lineales, que se pueden aplicar.

Uno de las operaciones lineales más importantes es la correlación, dado por:

$$O[u, v] = \sum_{(i,j) \in W} I[u + i, v + j] K[i, j], \quad \forall (u, v) \in I, \quad (2.3)$$

dónde $K \in \mathbb{R}^{W \times W}$ es el kernel y los elementos se conocen como los coeficientes del filtro. Para cada píxel de salida, la ventana de píxeles correspondiente de la imagen de entrada W se multiplica por el elemento K . El centro de la ventana y el núcleo se consideran coordenadas $(0, 0)$ e $i, j \in [-h, h] \subset \mathbb{Z} \times \mathbb{Z}$. La correlación frecuentemente se escribe de la forma $O = K \otimes I$. Y próximo a esta operación es la convolución $O = K * I$.

$$O[u, v] = \sum_{(i,j) \in W} I[u - i, v - j] K[i, j], \quad \forall (u, v) \in I, \quad (2.4)$$

Otra operación que se puede realizar es la detección de bordes, es uno de los métodos que se utiliza para descubrir un objeto en una escena. Se debe tener en cuenta la definición de dirección del borde que es la dirección de la tangente a un contorno que define el borde en el espacio bidimensional. Canny introduce una máscara que detecta la dirección del borde mediante la convolución de una función de detección lineal alineada normalmente a la dirección de la proyección paralelo a la detección de bordes [Peters, 2017].

Se utiliza el método de detección de bordes Laplaciano de Guassiano que trabaja con una suavización Gaussiana con un $K = 3 \times 3$ y $\sigma = 1.5$. Luego, realiza diferentes ecuaciones trigonométricas para hallar el ángulo de dirección y la intensidad. Después, descarta los puntos que no son considerados partes del borde para quedarse sólo con los potenciales bordes. Finalmente, se observa los píxeles que son mayores al umbral, estos serán considerados bordes. Por este motivo Canny es una de las técnicas más usadas, lo cual satisface los siguientes criterios:

- Baja tasa de error, detecta los bordes existentes sin tener falsos positivos.

- Localización eficiente, es decir, la distancia entre el borde detectado y el borde real es mínimo.

La morfología matemática es una clase de operadores espaciales no lineales. Cada píxel en la matriz de salida es una función de un subconjunto de píxeles en una región que rodea el píxel correspondiente en la imagen de entrada

$$O[u, v] = f(I[u + i, v + j], \forall(i, j) \in S, \forall(u, v) \in I, \quad (2.5)$$

dónde S es el elemento estructurante, una imagen binaria pequeña arbitraria. Para fines de implementación, este está incrustado en una ventana rectangular con longitudes de lados impares. El elemento estructurante es similar al kernel de convolución visto con anterioridad, excepto que el control para los píxeles de la vecindad se aplica la función f .

La operación morfológica que se utiliza más es la erosión y dilatación descritas como $O = I \ominus S$ y $O = I \oplus S$, respectivamente. La operación erosión se aplica en imágenes en grises, que permite obtener que las áreas cercanas al color blanco se adelgazarán, mientras que las zonas oscuras se percibirán mejor. Por otro lado, la dilatación permite quitar el ruido de la imagen, ya que expande las zonas de color negro.

2.2.2 Extracción de características

Las imágenes son simplemente grandes conjuntos de valores de píxeles, pero para las aplicaciones en el campo de robótica tienen demasiados datos y poca información. Por lo que se necesita responder las siguientes cuestiones: ¿cuál es la pose del objeto? ¿qué tipo de objeto es? ¿qué tan rápido se está moviendo? ¿qué tan rápido me estoy moviendo? y así sucesivamente. Las repuestas a tales preguntas son obtenidas a partir de la imagen y que es llamado características de la imagen. Las características son la esencia de la escena y la materia prima que se necesita para el control de un brazo manipulador [Corke, 2017].

Una definición está dado por el autor Jang, las características de una imagen están dados matemáticamente por la función f [Jang et al., 1991]:

$$f = \iint_{\text{image}} F(x, y, I(x, y)) dx dy, \quad (2.6)$$

dónde F es la función de mapeo que puede ser lineal o no lineal, dependiendo de la naturaleza de la característica en consideración. Además, (x, y) es la locación de la imagen y la intensidad del brillo está dado por $I(x, y)$.

Características de la Región

La segmentación de imágenes es el proceso de partición de las imágenes en regiones significativas. El objetivo es separar los píxeles que representan objetos de interés de todos los demás

píxeles en la escena. Este es uno de los enfoques más antiguos para entender la escena y, aunque conceptualmente directamente, es muy difícil en la práctica. Un requisito clave es la solidez, que es la forma en que el método se degrada a medida que se fuerza las suposiciones subyacentes, por ejemplo, cambiar la iluminación de la escena o la visualización de la escena [Corke, 2017].

La segmentación de imágenes se considera como tres subproblemas. La primera es la clasificación, que es un proceso de decisión aplicado a cada píxel que asigna el píxel a una de las clases C . Comúnmente usamos $C = 2$ que se conoce como clasificación binaria o binarización. Los píxeles se han clasificado como objeto ($c = 1$) o no-objeto ($c = 0$), que se muestran como píxeles blancos o negros, respectivamente. Las clasificaciones siempre son específicas para una tarea; por ejemplo, el objeto corresponde a píxeles brillantes, amarillos, rojos o en movimiento.

Si las regiones son homogéneas con respecto a algunas características como el brillo, el color o la textura. En la práctica, aceptamos que esta etapa es imperfecta y que los píxeles pueden clasificarse erróneamente; los siguientes pasos de procesamiento tendrán que lidiar con esto.

El segundo paso en el proceso de segmentación es la representación a la que se conectan los píxeles adyacentes de la misma clase desde conjuntos espaciales $S_1 \dots S_m$. Los conjuntos se pueden representar asignando una etiqueta de conjunto a cada píxel o mediante una lista de coordenadas de píxel que define el límite del conjunto conectado. En el tercer y último lugar, los conjuntos S_i se describen en términos de características escalares compactas o valores vectoriales tales como tamaño, posición y forma.

Características Circulares

Existe características circulares en algunos objetos que pueden ser halladas por la función de la transformada de Hough. Esta técnica se basa en la extracción de características para detectar círculos a pesar de las imperfecciones que pueda poseer. Esta técnica determina los parámetros de un círculo dependiendo del número de puntos que están dentro del perímetro conocido. Por cierto, un círculo tiene un radio R y un centro (a, b) que están descritos en las ecuaciones paramétricas:

$$x = a + R \cos(\theta) \quad (2.7)$$

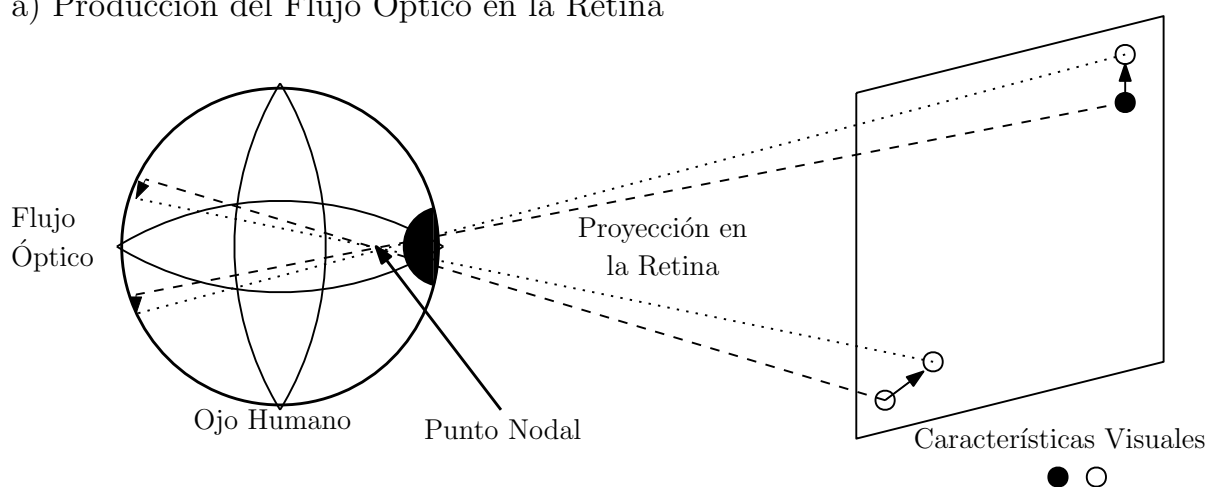
$$y = b + R \sin(\theta) \quad (2.8)$$

Después, se traza el perímetro del círculo con el centro (x, y) que realiza un barrido con ángulo θ por medio del rango de $0^\circ - 360^\circ$. Sin embargo, una imagen contiene muchos puntos donde algunos de ellos están dentro del perímetro del círculo, por lo que son almacenados en un círculo a, b, R . Así, los parámetros son introducidos en un vector acumulador que observa el número de repeticiones que posee, siendo retornados como círculos encontrados de la imagen.

2.3 Flujo óptico

Flujo óptico consta del cambio de la estructura de luz en la imagen [Raudies, 2013]; por ejemplo, en la Figura 2.5 se observa en la detección del flujo óptico en la retina, dónde dos objetos se mueven y que son dibujados en la retina con la dirección del movimiento de cada objeto. Por lo que, se puede obtener el movimiento de los objetos en dos *frames* consecutivos de la imagen, ya que se analiza la posición anterior de los píxeles con la imagen actual que pueden ser apreciadas en el inciso b de la Fig. 2.5. En este inciso se observa tres *frames*, lo cual se analiza el movimiento de rostro entre los *frames* 1 y 2, luego la traslación en otra dirección del rostro en los *frames* 2 y 3.

a) Producción del Flujo Óptico en la Retina



b) Detección del Flujo Óptico en el Plano de la Imagen

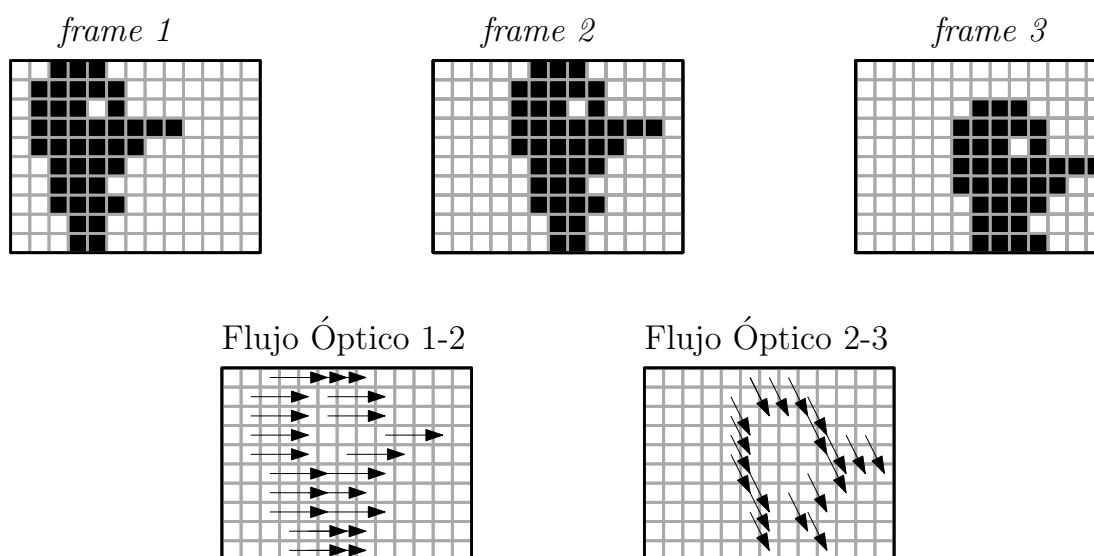


Figura 2.5: Reproducción del flujo óptico en la retina y las imágenes [Raudies, 2013].

Flujo óptico se basa en *frames* consecutivos, dónde cada píxel del primer *frame* es $I(x, y, t)$. Por lo cual, se debe hallar el desplazamiento del objeto (dx, dy) que se ha dado en instante de

tiempo de cada *frame* (dt).

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (2.9)$$

Se descubre dx, dy por medio de la aproximación de la serie de Taylor $f_x u + f_y v + f_t = 0$, en el cual f_x y f_y son la gradiente de la imagen. El principal problema es obtener u, v , por lo que se ha diseñado el método de Lucas-Kanade. Este método se basa en la división de la imagen original en pequeñas secciones, se asume que la velocidad es constante en cada sección v_x, v_y . Así que, se debe solucionar la ecuación básica del flujo óptico para las secciones i por el criterio de mínimos cuadrados.

$$\min \sum_i (f_{xi}u + f_{yi}v + f_t)^2 \quad (2.10)$$

$$\sum (f_{xi}u + f_{yi}v + f_t)f_{xi} = 0 \quad (2.11)$$

$$\sum (f_{xi}u + f_{yi}v + f_t)f_{yi} = 0 \quad (2.12)$$

$$\sum f_{xi}^2 u + \sum f_{xi} f_{yi} v = - \sum f_{xi} f_t \quad (2.13)$$

$$\sum f_{xi} f_{yi} u + \sum f_{yi}^2 v = - \sum f_{yi} f_t \quad (2.14)$$

Las ecuaciones anteriores pueden ser representadas matricialmente, por lo que, se debe tener en claro el concepto del método de Lucas-Kanade, representado como:

$$\begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} - \sum f_{xi} f_t \\ - \sum f_{yi} f_t \end{bmatrix} \quad (2.15)$$

Al emplear flujo óptico entre la imagen actual y la imagen anterior, se puede obtener el desplazamiento del movimiento (x, y) del objeto. Se observa que el flujo óptico implica una alta complejidad computacional, por este motivo se prefiere realizar los cálculos en los contornos del objeto, del cual se desea observar el movimiento.

El autor Rudies nos permite observar en la Figura 2.6, las restricciones empleadas en los métodos que estiman flujo óptico como $\vec{v} = (u, v)$ de una secuencia de imágenes en el espacio $f(x, y, t)$ [Raudies, 2013].

2.4 Estimación de Profundidad

Se tiene diferentes vertientes dependiendo del número de cámaras y de las configuraciones de las mismas. En esta investigación se va a centrar en la estimación de la profundidad por medio de imágenes monoculares. Por lo que, se utiliza diferentes técnicas para la estimación de profundidad.

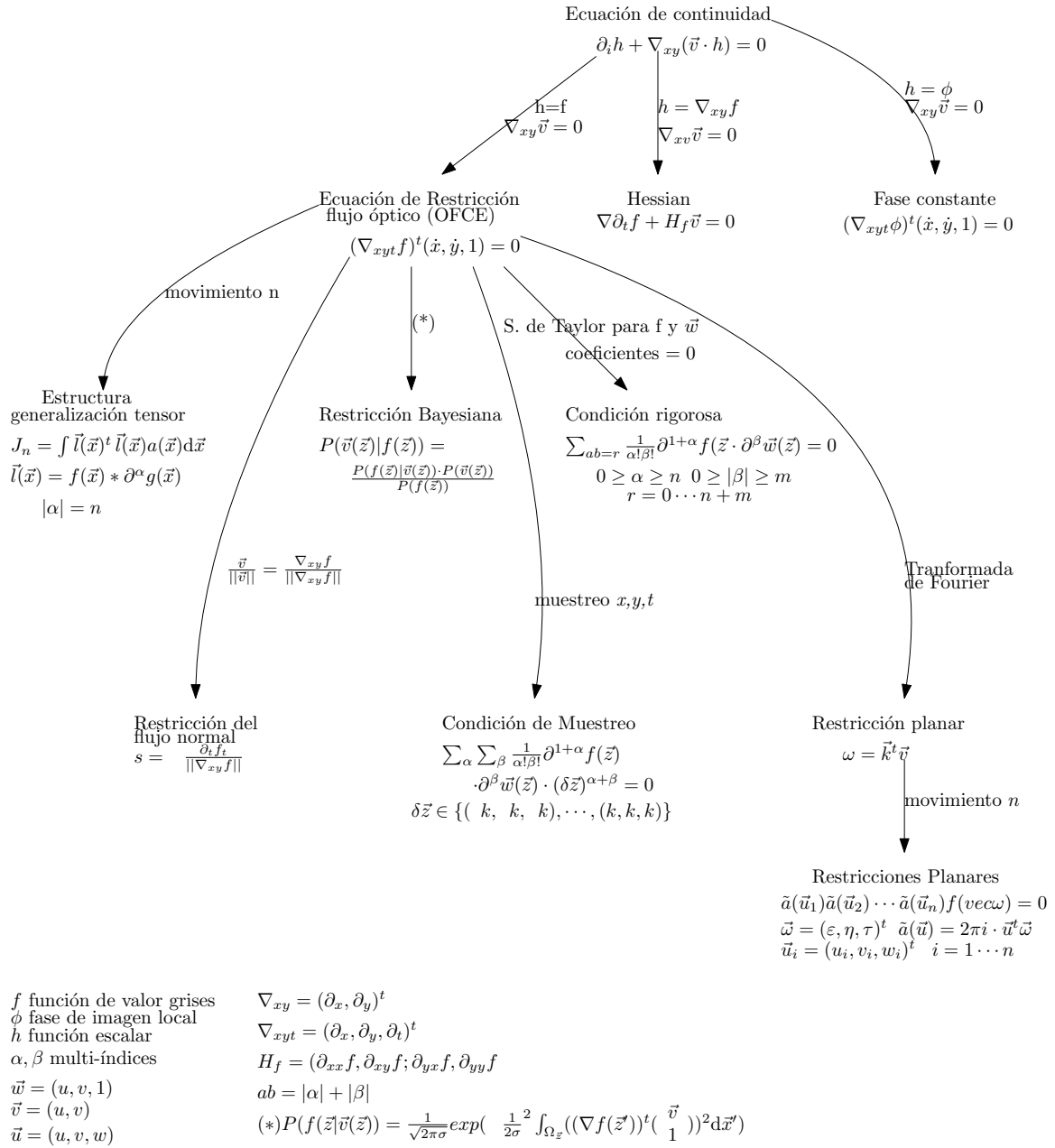


Figura 2.6: Taxonomía de las restricciones para la estimación del flujo óptico [Raudies, 2013].

La primera técnica usada se basa en filtros gaussianos que es la primera implementación del filtro de Bayes para espacios continuos. La idea original es la representación de las creencias por distribuciones normalizadas multivariantes, dadas por la ecuación 2.16

$$p(x) = \det(2\pi \sum)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \sum (x - \mu)\right\} \quad (2.16)$$

Esta densidad sobre la variable x es caracterizada por dos conjuntos de parámetros: La media μ y la covarianza \sum . La media es un vector que posee la misma dimensionalidad que el vector de estados x . La representación de una gaussiana por medio de la media y la covarianza es llamada *representación de momentos* [Thrun et al., 2005].

Después, surgió el **Filtro de Kalman (KF)** creado por Rudolph Emil Kalman en 1950, como una técnica para filtrar y predecir sistemas lineales. KF representa la creencia por la representación de momentos como: el tiempo t , la creencia es representada por la media μ_t y la covarianza \sum_t [Thrun et al., 2005]. Posteriormente la gaussiana tiene las siguientes propiedades:

1. La probabilidad del estado siguiente $p(x_t|u_t, x_{t-1})$ debe ser una función lineal con ruido añadido.

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad (2.17)$$

2. La probabilidad de *measurement* $p(z_t|x_t)$ debe ser argumentos lineales, con ruido añadido gaussiano:

$$z_t = C_t x_t + \delta_t \quad (2.18)$$

$$p(z_t|x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\} \quad (2.19)$$

3. Finalmente, la creencia inicial $bel(x_0)$ debe ser una distribución normal.

$$bel(x_0) = p(x_0) = \det(2\pi \sum_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_0 - \mu_0)^T \sum_0^{-1} (x_0 - \mu_0)\right\} \quad (2.20)$$

Para estimar la profundidad se aplica KF, en dónde se observa cada uno de las propiedades presentadas anteriormente y define la implementación de las etapas de extrapolación y actualización. Primero, se debe diseñar la representación para los estados por medio del algoritmo basado en píxeles (icónico) de profundidad del movimiento. Los estados son representados en un mapa de profundidad, donde se estima el valor en cada punto en la imagen actual. Para el enfoque basado en características, se estima la ubicación tridimensional de cada característica (en nuestro caso el borde) [Matthies et al., 1993].

Se estima y propaga un mapa de incertidumbre. Además, para el enfoque icónico, el ruido del *measurement* puede variar espacialmente debido al contraste local en la imagen. En el enfoque basado en característica, la precisión de las posiciones de los bordes también puede variar. Por lo tanto, para ambos métodos, la etapa de *measurement* inicial produce no solo la profundidad, sino también una varianza asociada.

La etapa de extrapolación para los dos enfoques comparte la misma ecuación de movimiento, pero difiere debido a la representación subyacente. Para el método icónico, el mapa se

deforma para predecir cómo se verá en el siguiente *frame*, y se lo re-muestrea para mantenerlo icónico. Para el método basado en características, la posición tridimensional de las características se extrapola.

Finalmente, el modelo anterior se puede usar para insertar conocimientos previos sobre la escena. En particular, las restricciones de suavidad (que requieren que los puntos cercanos tengan una disparidad similar) se pueden integrar fácilmente en el método icónico y se pueden usar para reducir la naturaleza ruidosa de las estimaciones de flujo. Para el enfoque de seguimiento de bordes, se podría utilizar la continuidad figurativa.

2.5 Visual Servoing

El término *visual servoing* fue acuñado por Hutchinson y se refiere al sistema de control de un robot con retroalimentación visual [Hutchinson et al., 1996]. Este sistema de control se basa en la extracción de características visuales como puntos, líneas y regiones que son usados para alinear un mecanismo manipulador o una herramienta con un objeto. Esta retroalimentación visual brinda un estado del ambiente. Por ejemplo, Shirie e Inoue usan *visual servoing* en un robot que sujeta un prisma cuadrado para colocarlo en una caja, para tener una mayor precisión; se basan en la extracción de bordes y de ajuste de líneas para determinar la posición y orientación de la caja [Hutchinson et al., 1996].

Visual servoing estudia las diversas formas, comenzando con tareas simples como sujetar y colocar cosas, hasta tareas en tiempo real, por ejemplo jugar ping-pong. La motivación principal es la incorporación de un sistema de visión en el sistema de control, ya que, existe una demanda creciente de sistemas robóticos flexibles.

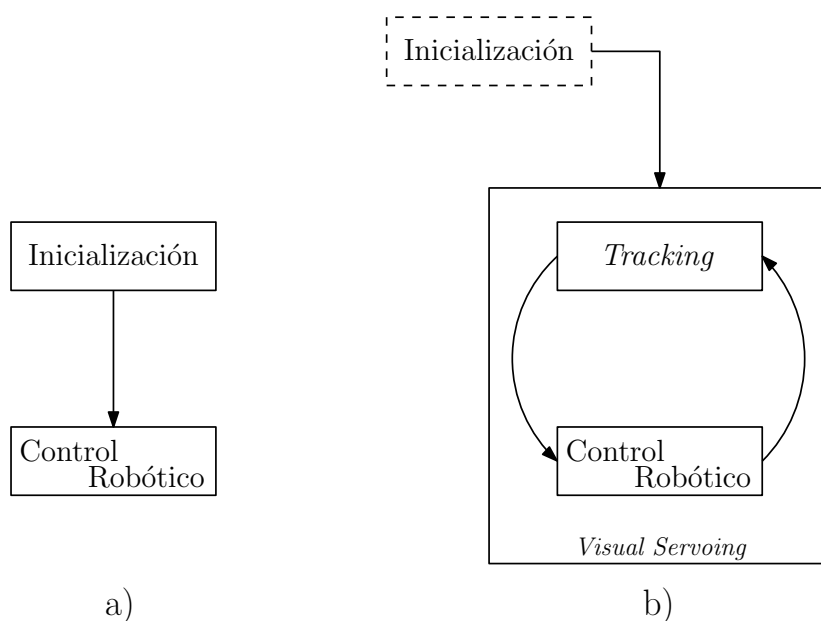


Figura 2.7: Vertientes fundamentales de los sistemas de control de brazos manipuladores [Kragic and Christensen, 2002].

Existen dos tipos de sistemas de control que son de lazo cerrado y de lazo abierto. La

Figura 2.7, se muestra en la imagen *a* es el sistema de control de lazo abierto; y el *b* es el sistema de control de lazo cerrado. Un sistema de control de lazo cerrado se usa el sensor de visión como subyacente, usualmente consiste en dos procesos: *Tracking* brinda una estimación continua y actualización de las características durante el movimiento del robot u objeto. El otro proceso es el sistema de control basado en la entrada sensorial que genera una secuencia de control. Este sistema requiere una inicialización automática que comúnmente incluye la segmentación del fondo de la imagen y reconocimiento del objeto. Por otro lado, el sistema de control de lazo abierto consiste en la inicialización que representa la extracción de las características que se emplea para generar la secuencia de control, no existe una interacción entre el robot y el ambiente luego de terminar la inicialización [Kragic and Christensen, 2002].

En la literatura, *visual servoing* es considerado como sistema de control visual de lazo cerrado. Por lo cual, desde este momento se va a hablar de un sistema de control de lazo cerrado basado en la información visual como *visual servoing*. Se tiene varias clasificaciones de las técnicas de sistemas de control visual basadas en: esquema de control, modelo del objetivo o simplemente en el error de la función cinemática.

En el campo de esquema de control se tiene dos esquemas fundamentales [Malis, 2002]:

- ***Direct visual servoing***: Se basa en calcular directamente la información visual como entrada de los sistemas dinámicos de control. Por ejemplo, Harrell usa este esquema para un brazo robótico de recolección de fruta con 2 GDL. Por medio de la información visual se determina la distancia para agarrar las frutas, es decir, directamente se usa la información visual para obtener la posición y orientación de las articulaciones del brazo manipulador [Corke, 1997].
- ***Dynamic look-and-move systems***: Este sistema realiza el control del robot en dos etapas, la primera es el sistema de visión que provee la entrada del controlador del brazo robótico, y la segunda etapa es la retroalimentación de las articulaciones para la estabilización interna del brazo manipulador.

En cambio, cuando se habla del modelo del objetivo se aborda si el modelo es conocido; sólo se tiene dos abordajes [Malis, 2002], como el anterior, que son:

- ***Model-based visual servoing***: Este abordaje se debe conocer el modelo tridimensional del objetivo. Es decir, se permite conocer las coordenadas del objetivo, cámara y posición actual. Teniendo estos parámetros se puede sacar 4 puntos referenciales del objetivo, lo cual nos permite obtener los parámetros de la cámara. De forma contraria, se necesita tener conocimiento de los parámetros de la cámara.
- ***Model-free visual servoing***: Este modelo desconoce el objetivo, por lo que la cámara toma una imagen del objetivo para que sea almacenado y sirva como un referencial. Luego, se saca una posición referencial entre la posición de la cámara respecto al objetivo. Lo cual, permite que se realice la estimación del desplazamiento del robot, por lo que se debe tener un vector para almacenar el error del control.

La última clasificación está basada en el error de la función cinemática de control que consta de las siguientes partes: *Visual-motor model estimation*, estimación de control y ge-

neración de control. Cada uno de estas partes serán descritas en las siguientes subsecciones [Kragic and Christensen, 2002].

2.5.1 Visual-Motor Model Estimation

En esta parte se realiza una diferenciación entre cinemática visual y control dinámico visual, el primer enfoque se centra en la realización del movimiento como respuesta de la información visual percibida. En cambio, el otro enfoque se centra en los efectos dinámicos que ocurren en un sistema. Por este motivo se clasifican en dos enfoques: sistema que se conoce a priori el *visual-motor model* y estimación de *visual-motor model* [Kragic and Christensen, 2002].

Como se observó anteriormente se tiene dos enfoques para la clasificación y además para el primer enfoque se sub-clasifica en la representación de la retroalimentación visual y el nivel de calibración entre la cámara y el *frame* del robot. Los primeros sistemas de *visual servoing* se basan en la calibración precisa de la cámara y realización de las tareas basadas en la posición. Sin embargo, el problema se da en el proceso de calibración por su complejidad. Por lo cual es propenso a errores e incluso imposible de realizar. Se empieza a preferir las tareas sin calibración exacta con conocimiento mínimo de la transformación entre la cámara y el *frame* del robot.

La clasificación para el enfoque en conocimiento a priori del *visual-motor model*, se basa en el modelo que representa la retroalimentación visual. Se tiene los siguientes abordajes que se basan en la posición, en la imagen y en un híbrido de los dos anteriores (llamado $2\frac{1}{2}D$).

Visual Servoing basado en Posiciones (PBVS) normalmente se conoce como control de un *3D servoing*, ya que realiza mediciones a la imagen para determinar la pose del objetivo con respecto a la cámara o algún *frame* del mundo en común. Además, el error entre la pose actual y la deseada del objetivo se define en el espacio de trabajo del robot (coordenadas cartesianas) [Kragic and Christensen, 2002, Corke, 1994] (Fig. 2.8).

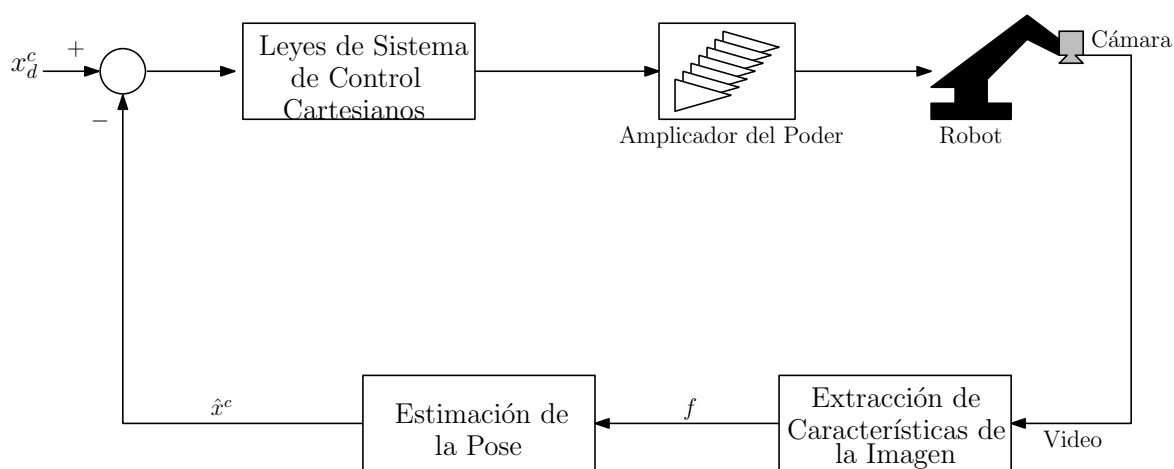


Figura 2.8: Estructura del *visual servoing* basado en posiciones (PBVS) [Corke, 1994].

La otra clasificación es un sistema visual basado en imagen, se aplica una estimación de la velocidad del robot a fin de moverse basándose en las características del plano de la imagen a un conjunto de posiciones deseadas. Para realizar este sistema se debe calcular la matriz Jacobiana

de la imagen, la cual representa la relación diferencial entre el *frame* de la escena y de la cámara [Corke, 1994, Kragic and Christensen, 2002] (Fig. 2.9).

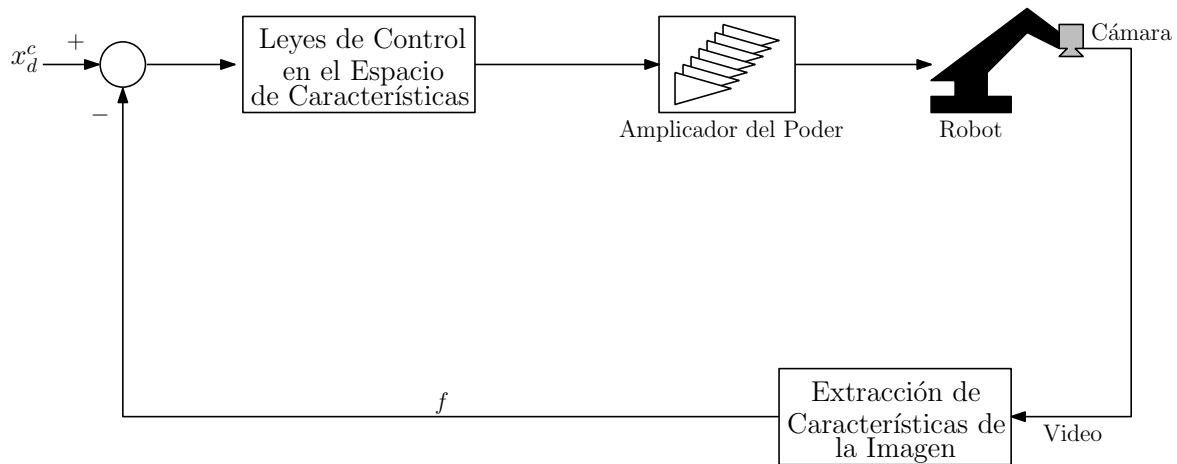


Figura 2.9: Estructura del control visual basado en imágenes (IBVS) [Corke, 1994].

El último enfoque es un híbrido, entre los abordajes anteriores como el empleo de PBVS, ya que no necesita de un modelo geométrico tridimensional del objetivo y además este sistema garantiza la convergencia de la ley de control en todo el espacio de trabajo que por lo contrario no lo hace *Visual Servoing* basado en Imágenes (IBVS). Este método se basa en la estimación del desplazamiento de la cámara entre la vista actual y deseada del objetivo. Por lo que, en cada iteración de estos dos puntos de vista, se estima la rotación y traslación de forma desacoplada [Corke and Hutchinson, 2001, Kragic and Christensen, 2002] [Rahmatillah et al., 2013].

2.5.2 Estimación de Control [Kragic and Christensen, 2002]

La estimación de control depende de la configuración de la cámara, el número de cámaras y las técnicas de procesamiento de imágenes. Además, se debe tomar en cuenta la posición y orientación del objetivo o las características, por lo que se debe definir la extracción de información visual. Por lo que, existe diferentes enfoques para realizar la estimación de control.

El primer enfoque son los sistemas monoculares, que emplean una cámara que se divide en dos configuraciones *eye-to-hand* y *eye-in-hand*. La primera configuración, se basa en una cámara global que observa el brazo manipulador y su espacio de trabajo, por lo que limita la visión del entorno del brazo. Este motivo, proporciona la configuración *eye-in-hand* que permite mayor alcance de visión y minimiza el tiempo de procesamiento para extraer la información visual. Sin embargo, se pierde una gran cantidad de información de la profundidad.

Además, existe el enfoque de visión binocular que consta de dos cámaras que proporciona información tridimensional completa del entorno. Este enfoque se basa en la disparidad que proporcionan una solución para la estimación de profundidad. El principal problema de este enfoque es la coincidencia de las características entre las imágenes de ambas cámaras, por lo que se proporciona dos métodos de solución:

- Coincidente por regiones correlacionados.

- Acorde con las características (esquinas y bordes) entre las imágenes.

La diferencia entre este enfoque y el anterior, es el consumo de tiempo computacional en cada iteración. Es decir, este enfoque permite estimar la profundidad de forma rápida, pero necesita de una mayor cantidad de memoria y tiempo computacional.

El otro enfoque son los sistemas de cámaras redundantes, es decir, múltiples cámaras. Este sistema proporciona información adicional que los anteriores enfoques, pero generalmente se necesita de una mayor cantidad de tiempo computacional, ya que las soluciones no son triviales. Por ejemplo, un sistema de visión triocular con dos cámaras en un sistema estéreo paralelo y una cámara global con visión vertical, observa mayor detalle de los diferentes movimientos del objetivo y el brazo manipulador para rectificar la trayectoria. Sin embargo, se debe observar la coincidencia entre las tres imágenes para dar una posición acorde a la información adicional de las tres cámaras.

En consecuencia, las cámaras redundantes se tiene diferentes problemas en el procesamiento de imágenes y el sistema de control visual robusto. Por lo que, se ha ideado los siguientes métodos de solución:

- Estimación en tiempo real de la pose del objetivo con respecto al sistema de coordenadas de la cámara.
- Implementación del sistema de coincidencia de control.

Generación de Control

Esta etapa está enlazada con *visual-motor model estimation*, porque el modelo afecta directamente a la velocidad de convergencia del sistema de control. Por otro lado, se debe tomar en cuenta los problemas de latencia y no linealidad de la cámara, el procesamiento de imágenes, la información sensorial dudosa [Kragic and Christensen, 2002].

Existe varios enfoques, el principal es el empleo de controladores proporcionales que necesita realizar varias pruebas para tener un buen desempeño en el controlador. Otro enfoque, modelar los *frames* como autómatas de estado finito que representan la ejecución de acciones, mientras que los eventos son las observaciones de cada acción. Este enfoque se basa en la entrada sensorial permitiendo la transición al siguiente estado, su principal problema es la velocidad de convergencia [Košecká et al., 1995].

2.6 Proceso de Control de Brazos Manipuladores

Para realizar el sistema de control de un brazo manipulador se debe seguir los siguientes pasos (Fig. 2.10):

1. Calcular los parámetros del brazo manipulador, dando los datos físicos del brazo, con lo que se realiza el sistema de referencia de los actuadores con respecto a la base.

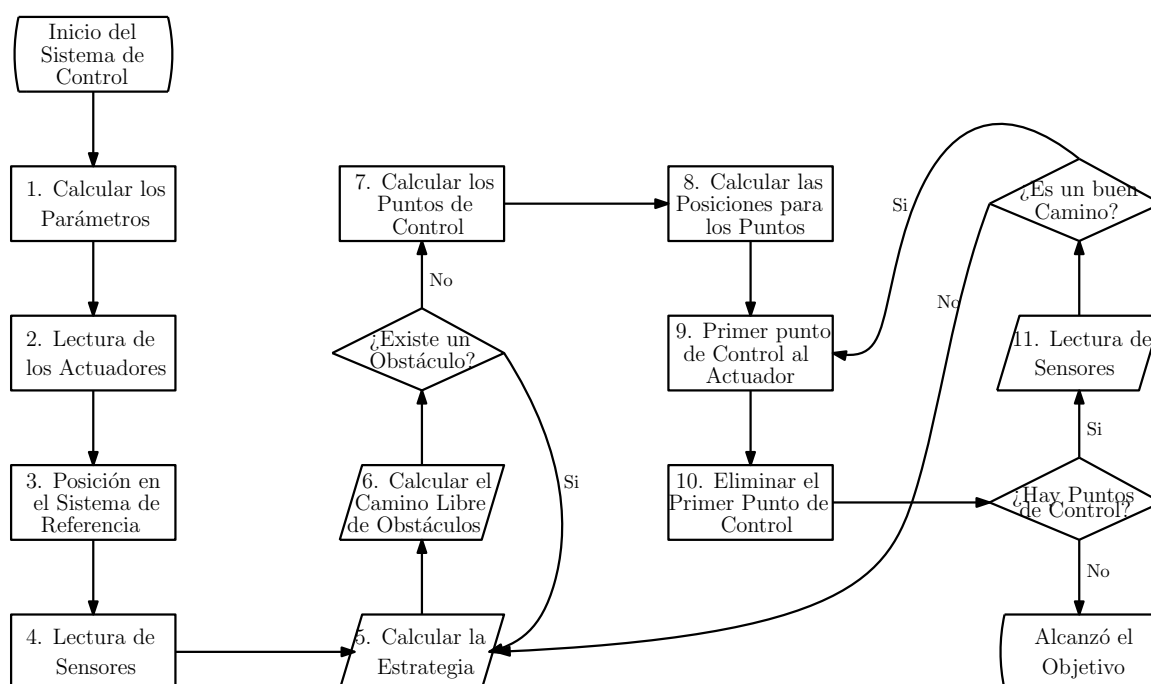


Figura 2.10: Diagrama de bloques para el control de un brazo manipulador.

2. Lectura de las posiciones de los actuadores de cada articulación, las cuales se toman como referencia inicial, para planificar la trayectoria del brazo al punto objetivo.
3. Posición actual en el sistema de referencia, es decir, que por medio del paso anterior y de matrices de transformación homogénea se dará la posición actual en el sistema de referencias, dado en el paso 1 (cinemática directa).
4. Lectura de sensores, proporciona la parte sensorial del brazo, que permite calcular la posición del objetivo con respecto al sistema de referencias del brazo y observar los obstáculos existentes. Posibilitando la observación de los posibles obstáculos en que están en la trayectoria para llegar al objetivo.
5. Realizar la estrategia para llegar al objetivo, en este paso se elige el camino más corto para llegar al objetivo teniendo en cuenta el paso 2 y 3.
6. Calcular un camino libre de obstáculos (para cada articulación), probando los posibles caminos tomando en cuenta la posición de los obstáculos en la parte 4, si está bloqueado por un obstáculo se regresa al paso anterior.
7. Calcular los puntos de control (posiciones de los efectores) dentro del camino libre, para tener un movimiento realista, se necesita suavizar el camino dado en el paso anterior que darán como resultado estos puntos de control.
8. En base a la cinemática inversa se calcula las posiciones deseadas de las articulaciones para llegar al objetivo (libre de obstáculos), es decir se realiza un arreglo de todas las posiciones de los puntos de control para llegar al objetivo.
9. Escribir progresivamente cada posición del arreglo de posiciones dado anteriormente, en el paso 8.
10. Eliminar el primer punto del arreglo dado en el paso 8.

11. Lectura de sensores se emplea para revisar si el camino es correcto. Es decir, si el camino planificado ya no es adecuado, volver al paso 5, en caso contrario volver al paso 9 mientras el arreglo siga lleno.

A continuación, se explicará los pasos más representativos mostrados anteriormente.

2.6.1 Parámetros de un Brazo Manipulador

Es posible obtener los parámetros de un brazo usando un sistema arbitrario para cada eslabón. En esta representación se toma en cuenta la transformación homogénea como un producto de cuatro transformaciones homogéneas [Spong et al., 2004a].

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (2.21)$$

Se resume estos sistemas por medio de la notación de D-H (Fig. 2.11), dada las siguientes instrucciones:

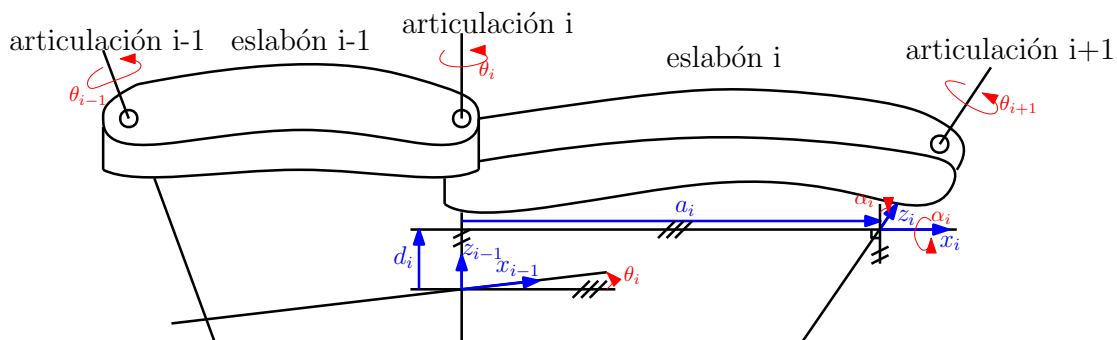


Figura 2.11: Representación de la notación D-H [Spong et al., 2004a].

1. Localizar y etiquetar los ejes de las articulaciones de z_0, \dots, z_{n-1} .
2. Establecer la base de las coordenadas como el conjunto del origen en el eje z_0 . Los ejes x_0 y y_0 son seleccionados convencionalmente con la regla de la mano derecha. Para $i = 1, \dots, n - 1$, luego ejecutar el paso 3 al 5.
3. Localizar el origen o_i donde la normal común es z_i y z_{i-1} intersecta en z_i . Si z_i intersecta a z_{i-1} localizado en o_i . Si z_i y z_{i-1} son paralelos, luego localiza o_i en cualquier posición conveniente a lo largo de z_i .
4. Establecer x_i a lo largo de la normal entre z_{i-1} y z_i de o_i , o en la dirección de la normal del plano $z_{i-1} - z_i$, si z_i y z_{i-1} se intersectan.
5. Establecer y_i con la regla de la mano derecha.
6. Establecer la herramienta final $o_n x_n y_n z_n$ asumiendo que la n-ésima articulación es rotacional. El conjunto $z_n = a$, a lo largo de la dirección z_{n-1} . Seguidamente, establecer el origen o_n convencionalmente al largo de z_n , preferentemente al centro de la herramienta

que el manipulador puede llegar. El conjunto $y_n = s$, en la dirección de la herramienta y el conjunto $x_n = n$ como $s \times a$.

7. Crear una tabla de los parámetros $a_i, d_i, \alpha_i, \theta_i$.

- a_i , es la distancia a lo largo de x_i de o_i de la intersección de los ejes x_i y z_{i-1} .
- d_i , es la distancia a lo largo de z_{i-1} de o_{i-1} de la intersección de los ejes x_i y z_{i-1} . Sólo existe esta variable si es una articulación prismática.
- α_i , es el ángulo entre z_{i-1} y z_i medido por x_i .
- θ_i , es el ángulo entre x_{i-1} y x_i medido por z_{i-1} . Sólo existe en articulaciones rotacionales.

8. Luego se forma las matrices de transformación homogénea A_i^{i-1} .

9. La forma $T_n^0 = A_1^0 \cdots A_n^{n-1}$, dada en la posición y la orientación expresada en la coordenada base.

2.6.2 Cinemática Directa

Un brazo manipulador puede ser considerado como una cadena cinemática formada por objetos rígidos entre sí mediante articulaciones, estableciendo un sistema de referencia fijo situado en la base del robot. Se describe la localización de cada uno de los eslabones con respecto a su sistema de referencia [Barrientos et al., 1997b].

La cinemática directa se reduce a encontrar la matriz homogénea de transformación T que relaciona la posición y la orientación del extremo del brazo manipulador respecto al sistema de referencia fijo. Para resolver la cinemática directa se debe encontrar la relación para conocer la localización espacial del extremo del robot a partir de los valores de sus coordenadas articulares. Se tiene las siguientes relaciones de acuerdo a la posición y la orientación de un brazo robótico de 6 GDL:

$$x = f_x(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.22)$$

$$y = f_y(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.23)$$

$$z = f_z(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.24)$$

$$\alpha = f_\alpha(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.25)$$

$$\beta = f_\beta(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.26)$$

$$\gamma = f_\gamma(q_1, q_2, q_3, q_4, q_5, q_6) \quad (2.27)$$

Se puede obtener la relación mediante los métodos geométricos. Por ejemplo, para un robot de 2 GDL:

$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \quad (2.28)$$

$$y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \quad (2.29)$$

Los brazos manipuladores de más grados de libertad se emplean matrices de transformación homogénea. La posición y la orientación relativa entre los sistemas asociados a los dos eslabones consecutivos del brazo robótico se suele denominar matriz A_i^{i-1} . Por ejemplo, se puede tener A_k^0 de las matrices resultantes del producto de las matrices A_i^{i-1} con i desde 1 hasta k , que representa de forma total o parcial la cadena cinemática que forma el robot. Es decir:

$$A_2^0 = A_1^0 A_2^1$$

En el ejemplo original se tiene la matriz T :

$$T = A_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5$$

Para representar el problema con D-H, se escoge los sistemas de coordenadas asociados a cada eslabón, es decir, se lleva de un eslabón a otro mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón. Por esto, sólo se puede ir del sistema $\{S_{i-1}\}$ al $\{S_i\}$, si estos sistemas están definidos como:

1. Rotación alrededor del eje z_{i-1} con un ángulo θ_i .
2. Traslación a lo largo de z_{i-1} en una distancia d_i (vector $(0, 0, d_i)$).
3. Traslación a lo largo de x_i en una distancia a_i (vector $(0, 0, a_i)$).
4. Rotación alrededor del eje x_i con un ángulo α_i .

Por lo que se describió anteriormente, estas matrices no son conmutativas. Las operaciones tienen que realizarse de acuerdo a:

$$A_i^{i-1} = T(z, \theta_i) T(0, 0, d_i) T(a_i, 0, 0) T(x, \alpha_i) \quad (2.30)$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.31)$$

$$= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

El ejemplo de la figura 2.12, está dado por la tabla 2.1 de D-H (extraído del capítulo de [Barrientos et al., 1997b])

En primer lugar, se debe localizar los sistemas de referencia de cada una de las articulaciones del robot. Se obtiene la Tabla 2.1 del brazo manipulador. Una vez calculado los

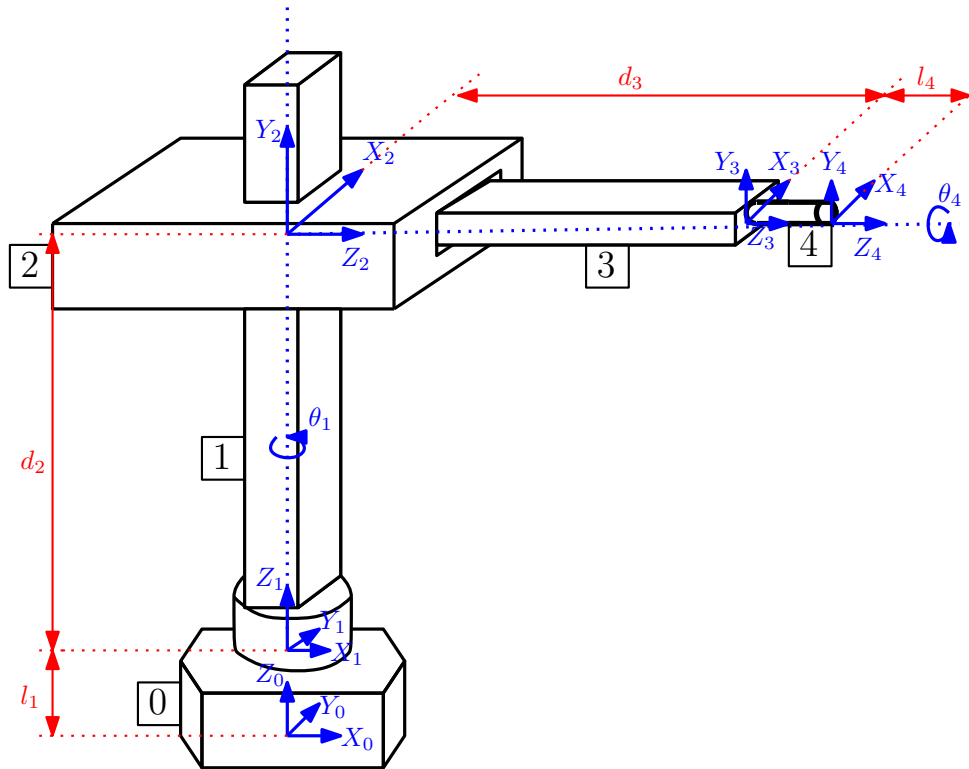


Figura 2.12: Brazo manipulador cilíndrico.

Tabla 2.1: Parámetros de D-H para el brazo manipulador cilíndrico (Fig. 2.12).

Articulación	θ	\mathbf{d}	\mathbf{a}	α
1	q_1	l_1	0	0
2	90	d_2	0	90
3	0	d_3	0	0
4	q_4	l_4	0	0

parámetros de cada eslabón, se calculan las matrices A , sustituyendo en las ecuaciones 2.33, 2.34, 2.35 y 2.36.

$$A_1^0 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.33)$$

$$A_2^1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.34)$$

$$A_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.35)$$

$$A_4^3 = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.36)$$

Así que se puede calcular la matriz T , es decir, la localización del sistema final con respecto al sistema de referencia de la base del brazo robótico.

$$T = A_1^0 A_2^1 A_3^2 A_4^3 \quad (2.37)$$

$$T = \begin{bmatrix} -S_1 C_4 & S_1 S_4 & C_1 & C_1(d_3 + l_4) \\ C_1 C_4 & -C_1 S_4 & S_1 & S_1(d_3 + l_4) \\ S_4 & C_4 & 0 & d_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.38)$$

2.6.3 Lectura de Sensores

Esta etapa se centra en obtener la información de los sensores, en la mayoría de casos es información visual. Por lo que, se debe tener en cuenta el tiempo computacional que toma limpiar los datos del sensor y la latencia que puede existir en tiempo real. Además, de los posibles errores de calibración o propios del sensor que afectan a la información que se está leyendo.

2.6.4 Estrategia para la siguiente posición

La estrategia depende de la tarea que va a desarrollar el brazo robótico para elegir la estrategia correspondiente. En otras palabras, los puntos que debe seguir el brazo manipulador para realizar una determinada tarea.

Por ejemplo, si el brazo manipulador está destinado a tareas de rescate. Las víctimas podrían estar atrapadas, entonces el brazo realiza un barrido del espacio para buscar las víctimas, de arriba-abajo y de derecha-izquierda, observando si hay un orificio, por el cual se envía el brazo para entrar en la abertura y observe si hay alguna víctima.

2.6.5 Calcular camino libre de obstáculos y puntos de control

Luego de ver los caminos deseados, para que el brazo llegue a una posición, se debe verificar por medio de su percepción si el camino está libre de obstáculos. Además, luego de elegir el mejor camino libre de obstáculos, hay que obtener los puntos de control. Estos puntos son fundamentales, ya que, si se envía todos los ángulos para cada efector en un mismo instante,

este podría golpearse con un objeto de su ambiente. Una de las formas para tratar este tipo de problemas es calculando los puntos de control, de tal forma que se realiza el camino poco a poco y que en un determinado tiempo pueda obtener los datos de percepción y poder asegurar que es un buen camino.

2.6.6 Cinemática Inversa

El problema de la cinemática inversa consiste en encontrar valores que adopten las coordenadas articulares del robot $q = [q_1, q_2, \dots, q_n]^T$, para que su extremo se posicione y se oriente en una determinada localización espacial [Barrientos et al., 1997b].

Este problema necesita de cálculos complejos para obtener una solución, pero se implementa por medio de algunos procedimientos genéricos para dar una solución. Por lo cual, a partir de los parámetros de D-H se puede obtener los valores articulares.

Para resolver el problema se debe encontrar una solución cerrada. Es decir, encontrar una relación matemática explícita de la forma:

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma) \quad (2.39)$$

$$k = 1 \dots n(GDL) \quad (2.40)$$

Las ventajas para usar este método son:

- Se debe resolver este problema en tiempo real. Por lo cual, una solución de tipo iterativo no garantiza tener la solución en el momento adecuado.
- Para la cinemática inversa la solución no es única. Existen diferentes soluciones para una misma orientación. Por lo que, una solución cerrada permite incluir determinadas reglas o restricciones que aseguran que la solución obtenida sea la más adecuada entre las posibles.

Se propone revisar las características de esta cinemática a partir de T_N^0 , posee 16 elementos [Alsina, 2014]. De estos elementos, 4 son triviales y 9 son la matriz de rotación sujeto a 6 restricciones. Al mapear esta matriz, podemos darnos cuenta que hay infinitas soluciones.

Se tiene la existencia de una solución, si:

- En el espacio de trabajo existe especificaciones para la matriz T .
- En el espacio de trabajo es alcanzable por la matriz de orientación.
- Por último, el espacio de trabajo tiene que ser manipulable.

Se tiene múltiples soluciones de acuerdo al número de articulaciones:

- $N = 6$, se tiene seis ecuaciones y seis incógnitas. Existe un número finito de soluciones.
- $N < 6$, se tiene de igual manera seis ecuaciones y menos de seis incógnitas. La solución está en un sub-espacio del espacio en 3D.
- $N > 6$, para este caso se tiene infinitas soluciones.

Resolución del dilema de cinemática inversa a partir de métodos geométricos

La solución del problema de la cinemática inversa para un brazo manipulador se puede resolver por métodos geométricos [Craig, 2006]. Para esto, se debe descomponer la geometría espacial del brazo en varios problemas de geometría plana. Los brazos manipuladores donde $\alpha_i = 0$ y $\alpha_i = \pm 90$; por ejemplo, en un brazo de 3 Grados de Libertad, se puede realizar directamente la geometría plana para encontrar una solución.

Por ejemplo, la Figura 2.13 muestra el triángulo formado por L_1 , L_2 y la línea que une el origen de la trama $\{0\}$ con el trama $\{3\}$. Las líneas punteadas son otras configuraciones para llegar a la misma trama. Si se aplica la ley de los cosenos, para resolver θ_2 :

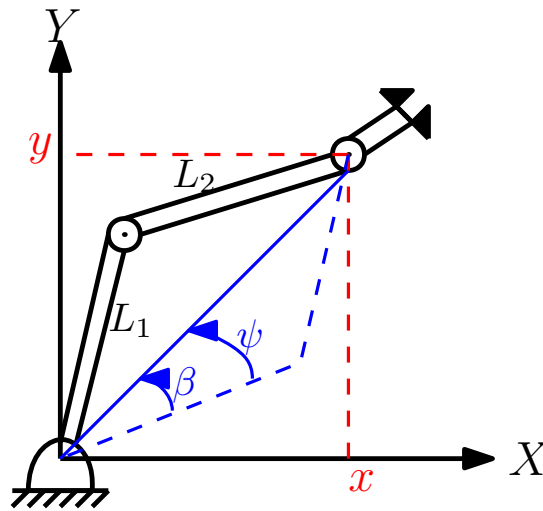


Figura 2.13: Geometría plana asociada al brazo manipulador de 3 GDL [Craig, 2006].

$$x^2 + y^2 = (L_1)^2 + (L_2)^2 - 2L_1L_2 \cos(180 + \theta_2) \quad (2.41)$$

Reemplazando, $\cos(180 + \theta_2)$ como $-\cos(\theta_2)$ y despejando \cos , se obtiene:

$$C_2 = \frac{x^2 + y^2 - (L_1)^2 - (L_2)^2}{2L_1L_2} \quad (2.42)$$

Se puede asegurar que existe una solución observando que la distancia del punto de destino es $(\sqrt{x^2 + y^2}) \leq L_1 + L_2$. Esta condición no satisface si el punto de destino está fuera del alcance del brazo manipulador.

Para despejar θ_1 se encuentra determinado por los ángulos ψ y β según la Fig. 2.13. Para hallar β se usa la función arco tangente de dos argumentos. Luego se aplica la ley de cosenos para encontrar ψ .

$$\cos \psi = \frac{x^2 + y^2 + (L_1)^2 - (L_2)^2}{2L_1\sqrt{x^2 + y^2}} \quad (2.43)$$

Para hallar θ_1 se debe aplicar las fórmulas de derivación sobre un rango de variables que preserve la geometría, es decir:

$$\theta_1 = \beta \pm \psi \quad (2.44)$$

Se usa el signo positivo, si $\theta_2 < 0$ y el signo negativo para el caso contrario. Además, se puede hablar del tercer ángulo:

$$\theta_1 + \theta_2 + \theta_3 = \phi \quad (2.45)$$

Resolución del dilema de cinemática inversa a partir de la matriz de transformación homogénea

Se puede obtener una solución a partir del conocimiento del modelo del brazo manipulador (cinemática directa). Sin embargo, a veces resolver la inversa de las matrices de transformación homogénea es no trivial y su complejidad se eleva [Barrientos et al., 1997b].

Por ejemplo, se resuelve un brazo manipulador con 3GDL que se observa en la Figura 2.14 y su modelo está dado por la Tabla 2.2. La matriz que se obtiene de este brazo es dada por $T = A_1^0 A_2^1 A_3^2$, en consecuencia, la inversa es:

$$(A_1^0)^{-1}T = A_2^1 A_3^2 \quad (2.46)$$

$$(A_2^1)^{-1}(A_1^0)^{-1}T = A_3^2 \quad (2.47)$$

Tabla 2.2: Parámetros de D-H del brazo manipulador de la Fig. 2.14.

Articulación	θ	\mathbf{d}	\mathbf{a}	α
1	q_1	l_1	0	90
2	q_2	0	0	-90
3	0	q_3	0	0

Para poder aplicar este procedimiento, es necesario observar la inversa de las matrices. Se debe considerar lo siguiente:

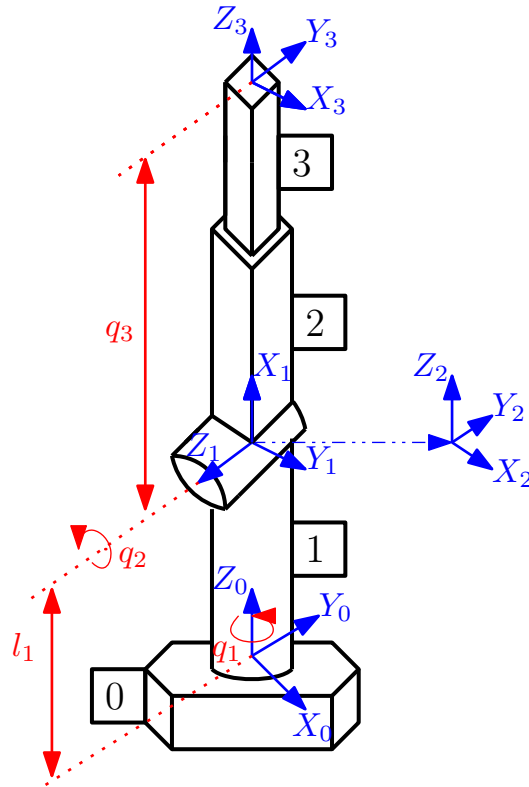


Figura 2.14: Sistema de referencias del brazo manipulador de 3 GDL [Barrientos et al., 1997b].

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T p \\ o_x & o_y & o_z & -o^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.48)$$

Se puede observar la inversa de las matrices presentadas anteriormente, como:

$$(A_1^0)^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.49)$$

$$(A_2^1)^{-1} = \begin{bmatrix} C_2 & S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.50)$$

$$(A_3^2)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.51)$$

$$(2.52)$$

Por lo tanto, usando las primeras ecuaciones se tiene que:

$$(A_1^0)^{-1}T_3^0 = A_2^1A_3^2 \quad (2.53)$$

$$= \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.54)$$

$$= \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.55)$$

$$= \begin{bmatrix} C_2 & 0 & -S_2 & -S_2q_3 \\ S_2 & 0 & C_2 & C_2q_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.56)$$

Establecido anteriormente las relaciones, se desea saber que la expresión q_1 en función a las constantes, es decir $q_1 = \arctan(\frac{p_y}{p_x})$. Luego, hallar q_2 :

$$(A_2^1)^{-1}(A_1^0)^{-1}T = A_3^2 \quad (2.57)$$

$$= \begin{bmatrix} C_2 & S_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.58)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.59)$$

Despejando la $\tan(q_2)$ como $-\frac{C_1p_x+S_1p_y}{(p_z-l_1)}$, considerando que $C_1p_x+S_1p_y = \sqrt{(p_x)^2 + (p_y)^2}$. Nos queda $q_2 = \arctan \frac{\sqrt{(p_x)^2+(p_y)^2}}{l_1-p_z}$.

Por lo tanto, podemos obtener los valores de las tres articulaciones:

$$q_1 = \arctan\left(\frac{p_y}{p_x}\right) \quad (2.60)$$

$$q_2 = \arctan \frac{\sqrt{(p_x)^2 + (p_y)^2}}{l_1 - p_z} \quad (2.61)$$

$$q_3 = C_2(p_z - l_1) - S_2\sqrt{(p_x)^2 + (p_y)^2} \quad (2.62)$$

Resolución del dilema de cinemática inversa a partir del desacoplo cinemático

Este método no solo trabaja con las tres primeras articulaciones del robot, sino también la posición del extremo del robot en las coordenadas (p_x, p_y, p_z) [Barrientos et al., 1997b].

En general, para un brazo robótico no solo basta con posicionar el extremo del brazo manipulador en un punto del espacio, sino también conseguir que la garra este en una orientación y una posición determinada.

Se enfatiza que, para realizar este método, se debe realizar tres pasos [Craig, 2006]:

1. Se debe encontrar q_1, q_2, q_3 de manera que el centro de la muñeca o_c son las coordenadas dadas por:

$$o_c^0 = o - d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.63)$$

2. Usando las variables de las articulaciones determinadas en el paso 1, se evalúa R_3^0 .
3. Se debe encontrar un conjunto de ángulos de Euler, correspondientes a la matriz de rotación.

$$R_6^3 = (R_3^0)^{-1} R = (R_3^0)^T R. \quad (2.64)$$

2.7 Consideraciones Finales

Tomando en cuenta las teorías presentadas con anterioridad, para este trabajo de investigación se tomará la representación en coordenadas cartesianas por medio de matrices homogéneas para mostrar la rotación y traslación del objeto y del brazo manipulador.

Por otro lado, el sistema de visión que se va tomar es el control de lazo cerrado, ya que el objetivo principal realiza el seguimiento de un objeto por lo que se necesita un sistema de control con retroalimentación en cada instante de tiempo. Además, el modelo que se usa es el *model-free visual servoing* que consta de varias etapas, que se optarán por las técnicas según sus etapas como se muestra a continuación:

- *Visual-motor model estimation*, se va empleará el método PBVS que consta en determinar la pose del objetivo con las mediciones de la imagen con respecto a la posición de la cámara en el mundo real.
- Estimación de control, se enfocará en un sistema monocular ya que el brazo manipulador que se modela en esta investigación consta de una cámara; por lo que se ha elegido una configuración *eye-in-hand* por la posición de la cámara que está localizada en la garra del brazo manipulador.
- Generación de control, se empleará controladores proporcionales por la rapidez de convergencia que se necesita al ser un sistema en tiempo real.

Al referirnos a la estimación de profundidad, ya que se mencionó en el anterior párrafo el sistema que se investiga consta de una cámara, por lo que la profundidad se deberá observar métodos para su estimación. Es por este motivo que se va a emplear KF. También, se debe tener en cuenta que se debe observar la traslación y rotación que se ha realizado el objetivo por lo que se usará el flujo óptico para obtener el movimiento del objetivo.

Capítulo 3

Estado del Arte

Este capítulo recopila investigaciones de los diferentes científicos de las diversas áreas que constituyen en el desarrollo de un sistema de control. Primero, se analizará los trabajos publicados del área de control visual para brazos manipuladores, luego, se observará las técnicas de modelado utilizadas en la literatura. Después, se analizará las diferentes técnicas basadas en flujo óptico empleadas para el seguimiento de objetos. Seguidamente, se observa las diferentes técnicas para la estimación de profundidad. Por último, se examinará las diferentes propuestas de cinemática inversa para un brazo manipulador.

3.1 *Visual Servoing*

Un sistema de control visual se refiere al sistema de control con retroalimentación visual de un brazo manipulador. Este sistema se basa en la extracción de características visuales para alinear un brazo manipulador. La mayoría de estos sistemas son utilizados con un sistema de visión estéreo, ya que requiere de la información tridimensional del objeto que se desea alcanzar por el brazo manipulador.

La principal desventaja de la visión estéreo es el costo computacional que requiere para realizar un *matching* entre las imágenes, además que el brazo manipulador tendría que estar siempre en la misma perspectiva con la cámara. Por este motivo, algunos autores han realizado el control visual con una cámara anclada en la garra del brazo robótico (*eye-in-hand*).

A continuación, podemos observar las diferentes investigaciones en este campo en los últimos años. Por ejemplo, el autor Wang y sus colegas realizan una investigación en brazos manipuladores simulados utilizando el esquema IBVS, lo cual proponen realizar un suavizado adaptativo inverso para la compensación del efecto *backlash*, ángulo de giro negativo al ángulo de giro original del eje de la articulación [Wang et al., 2016]. Abadianzadeh y sus colegas, proponen un nuevo control visual para brazos manipuladores aplicando controladores *fuzzy* híbridos que obtienen su entrada por una *kinect*. El primer controlador se basa en un modelo inverso *fuzzy* que aproxima el modelo inverso real del robot; y el segundo controlador es designado para suprimir la salida del primer controlador para ir al área designada [Abadianzadeh et al., 2016].

De manera similar, los autores Bueno-López y Arteaga-Pérez aplican una estrategia *fuzzy*

de control fusionada con control visual tridimensional. Se realiza la diferencia de la trayectoria deseada con la trayectoria obtenida de la visión estéreo, que dará la entrada al controlador *fuzzy* como señal de error. La salida del controlador será aplicada en el controlador del robot [Bueno-López and Arteaga-Pérez, 2011].

Los autores Chi-Ying et. al. aplican la composición virtual del modelo de la cámara y un espacio cartesiano visual tridimensional para calcular la matriz Jacobiana de la imagen para el control del seguimiento visual [Lin et al., 2016]. Por otro lado, el autor Cai y sus colegas, realizan un nuevo modelo de cámara estéreo en un brazo manipulador de 6GDL que emplea cámaras ortogonales virtuales para la construcción de un mapa de posición 6D, teniendo en cuenta que cada componente de este vector es linealmente independiente que permite hallar la matriz Jacobiana de la imagen. Además, se realiza el desacoplo de la diagonal de la matriz para el control del movimiento rotacional y traslación [Cai et al., 2016].

Los autores Pérez, Reza y Pomares utilizan información visual y realizan una optimización en el movimiento de las articulaciones con una compensación caótica que se ha aplicado en un brazo manipulador de 4GDL en diferentes escenarios [Alepuz et al., 2016]. Por otro lado, el autor Zhang y sus colegas, realizan un modelo cinemático de los robots por medio de *Finite Element Method* (FEM), que realiza una predicción de la matriz Jacobiana del robot. Por lo que, primero realiza un control en el espacio de articulaciones y luego una estrategia basada en la retroalimentación de cámaras infrarrojas que es desarrollado para obtener una corrección de la posición de la garra [Zhang et al., 2017].

Además, últimamente se utiliza la tecnología *Field Programmable Gate Array* (FPGA) que permite realizar implementaciones para controladores visuales en un chip. Por lo que el autor Alabdo y sus colegas al. utilizan métodos de compensación en comportamiento caótico [Alabdo et al., 2016] implementado en FPGA para realizar el seguimiento de imágenes. Otro caso, integración de entrada de visión tridimensional y el sistema de control de movimiento del robot en FPGA. Lo cual, en este chip se implementan las funciones de filtración, estadísticas de imágenes, morfología binaria, análisis de objetos binarios, calcular posición de objetos en 3D, cinemática inversa, descripción de generación de velocidad, conteo de retroalimentación y retroalimentación de control de posición de múltiples ejes [Chen et al., 2015].

Los autores Ching-Long, Jung-Hsien y Chi-Jen han implementado en FPGA el procesamiento de la visión y en otro chip realizan un control PID del robot [Shih et al., 2017]. En cambio, los autores Leite y Lizarralde realizan el sistema de control particionado que desarrollan un movimiento en el flujo óptico de la cámara y un movimiento bidimensional restringidos en el plano ortogonal de estos dos sub-sistemas. Por lo que, los autores desarrollan una estrategia de control en cascada basada en un método adaptativo indirecto/directo [Leite and Lizarralde, 2016].

3.2 Técnicas de Modelado de Brazos Manipuladores

El principal modelado es la representación geométrica que representa el sistema referencial de los *frames* para cada enlace. El *frame* se determina para cada enlace individual, pero sin realizar conexión con el siguiente enlace. Este modelado está diseñado para separar las matrices de traslación y rotación [Ajwad et al., 2014], es decir:

$$p^A = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.1)$$

$$R^x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C & -S \\ 0 & S & C \end{bmatrix} R^y = \begin{bmatrix} C & 0 & S \\ 0 & 1 & 0 \\ -S & 0 & C \end{bmatrix} R^z = \begin{bmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

La cinemática directa es la multiplicación de las transformaciones de cada enlace en una secuencia, es decir, desde la articulación de la base a la herramienta [Craig, 2005]. A pesar de ser una representación simple se utiliza mucho tiempo computacional, ya que consiste en una multiplicación intensiva de matrices para cada enlace.

Por lo que, surge la representación de Denavit-Hartenberg [Kamaliya and Patel, 2016], dónde cada enlace es considerado un cuerpo rígido. Luego, se asigna a cada enlace un sistema de referencia que está compuesto de cuatro variables y que están dadas según el tipo de articulación [Spong et al., 2004a]. Esta representación está compuesta por las matrices de transformación homogénea (Ecuación 3.3) que describen la posición y la orientación relativa de los sistemas referenciales.

$$T_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Esta es la representación más usada, por la versatilidad que nos propone trabajar con articulaciones prismáticas y rotacionales. Sólo se debe variar la tabla de la notación de D-H y sin realizar muchos cambios en la implementación. Por lo que, el autor Vihonen y et. al. aplican D-H con la función de sensores como acelerómetros lineales y giroscopios [Vihonen et al., 2016].

Otra representación está basada en la teoría del *screw*. Esta teoría fue desarrollada por Robert S. Ball en 1876 que emplea el teorema de Chasles de la cinemática y los mecanismos estáticos [Ball, 1900]. Este teorema describe el movimiento de un cuerpo rígido de acuerdo con la rotación y la traslación a lo largo de una línea llamada eje *screw* de movimiento (Fig. 3.1) [Ajwad et al., 2014].

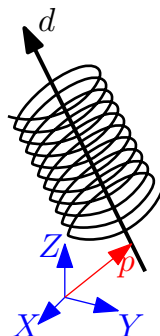


Figura 3.1: Representación del eje *screw* de Movimiento [Ajwad et al., 2014].

Se emplea el álgebra de matrices para describir el movimiento de forma matemática como se presenta a continuación [Bruyninckx and Schutter, 2001]:

$$T_i^{i-1} = \begin{bmatrix} R(\theta, d) & \frac{\theta(kd+(1-R(\theta d))p)}{2\pi} \\ 0 & 1 \end{bmatrix} \quad (3.4)$$

El movimiento *screw* puede ser expresado por cuaterniones simples y duales. Este es un método eficiente y compacto para describir el desplazamiento *screw*, sin embargo, surge el problema de obtener las referencias de cada enlace y la complejidad de tener múltiples enlaces [Ding et al., 2014].

De igual manera, existe la técnica de resolver los problemas de la notación D-H para los casos de líneas paralelas cercanas. Esta representación es llamada notación de Hayati-Roberts (H-R), los segmentos lineales son representados por $L_{hr}(e_x, e_y, l_x, l_y)$. Los parámetros e_x y e_y son X y Y de los componentes del vector unitario de dirección e en una línea. De esta forma se elimina la necesidad del componente Z del vector de dirección, ya que está representado como $e_z = (1 - e_x^2 - e_y^2)^{1/2}$. En cambio, los parámetros l_x y l_y son las coordenadas del punto de intersección de la línea con el plano a través del origen del *frame* de referencia al mundo [Ajwad et al., 2014].

Después de observar la convención de H-R dónde los ejes de los dos enlaces son paralelos. Los ángulos *roll* (γ) y *pitch* (β), se muestran en la Figura 3.2. Por lo que, la traslación y la rotación están denotadas por:

$$T_{i-1}^i = R(Z, \alpha^{i-1}) T(X, d^{i-1}) R(Y, \gamma^i) R(X, \alpha \beta^i) \quad (3.5)$$

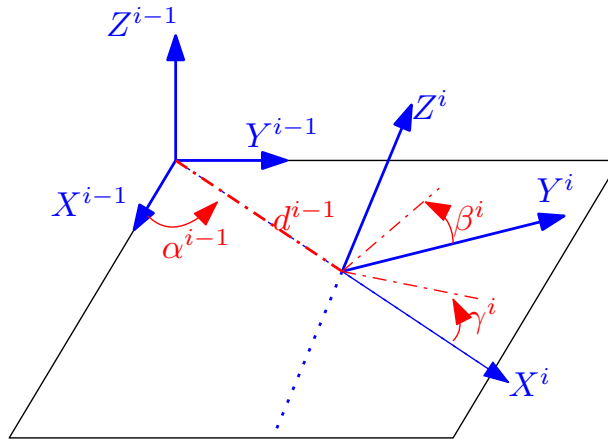


Figura 3.2: Convención de Hayati-Roberts [Ajwad et al., 2014].

Otra representación es la de Veitschegger y su colega que proponen usar la rotación alrededor del eje Y , concatenando con los cuatro parámetros de Denavit-Hartenberg. La principal desventaja surge en el cálculo cinemático [Veitschegger and Wu, 1987].

Los autores Beliflore y Benedetto aplican la teoría de grafos para representar la estructura cinemática redundante con 1 GDL para cada articulación. En otras palabras, cada articulación es un nodo y las uniones son los enlaces de cada articulación. Sin embargo, existe un grave problema con brazos manipuladores con estructura paralela [Belfiore and Benedetto, 2016].

La nueva propuesta de notación es dada por Thomas et. al. que usan la notación extendida de Denavit-Hartenberg, donde se aumenta un parámetro β [Thomas et al., 2002]. Esta variable es empleada cuando una articulación tiene más de 2 GDL. Por ejemplo, las articulaciones esféricas tienen los parámetros θ_i , β_i y α_i para moverse en los tres ejes coordenados. De igual manera, se utiliza la representación de Belifiore y Benedetto para modelar la estructura de brazos manipuladores [Belfiore and Benedetto, 2016].

Por otro lado, el autor Peng Qi y sus colegas realizan un modelado para brazos manipuladores con deformaciones, es decir aprovecha la rotación secuencial; por lo que, introduce una metodología de modelado unificado basado en el concepto de rotaciones simultáneas y el uso de la fórmula de rotación de Rodríguez [Qi et al., 2014].

Se ha elaborado una nueva propuesta para brazos manipuladores con uniones paralelas y articulaciones esféricas. Sin embargo, las articulaciones seriales se emplean los parámetros de Denavit-Hartenberg, por lo que al aumentar las variables incrementa la complejidad de la solución.

Para terminar, se elige la notación tomando en cuenta el tipo de articulaciones y la estructura del brazo manipulador que se desea modelar. Por lo que, en esta investigación se ha centrado en el uso de la representación de D-H, ya que la estructura del brazo robótico consta de articulaciones rotacionales.

3.3 Flujo Óptico

Flujo óptico es un campo denso de vectores de desplazamiento que definen la traslación para cada píxel en una región. Pero, existe la restricción computacional del brillo, ya que se asume que el brillo es constante para cada píxel en *frames* consecutivos [Horn and Schunck, 1981]. Flujo óptico comúnmente emplea características basado en segmentos para realizar el seguimiento de objetivos [Yilmaz et al., 2006].

Comúnmente se emplea el método de la gradiente basada en el flujo óptico en cada nivel del píxel. El autor Thota y sus colegas expresan matemáticamente la intensidad de cada píxel trasladado de un *frame* al siguiente [Thota et al., 2013]:

$$I(\vec{x}, t) = I(\vec{x} + \vec{u}, t + 1) \quad (3.6)$$

La intensidad de la imagen es denotado por $I(\vec{x}, t)$, que puede ser interpretada como una función del espacio $\vec{x} = (x, y)^T$, donde el tiempo t y la velocidad bidimensional $\vec{u} = (u_1, u_2)^T$. Después, se expande utilizando la regla de la mano de la ecuación 3.7 y expresado en términos de la serie de Taylor de primer orden.

$$I_x \cdot u + I_y \cdot v + I_t = 0 \quad (3.7)$$

dónde, I_x , I_y e I_t son las derivadas parciales de la intensidad con respecto a los elementos x , y y t respectivamente. Por otro lado, $u = \frac{dx}{dt}$ y $v = \frac{dy}{dt}$ son los componentes direccionales de

x y y .

La desventaja de los métodos de la gradiente es que no pueden obtener la solución completa por el problema de la apertura, es decir, un componente normal paralelo a la dirección de la gradiente, está determinado por los componentes perpendiculares tangentes a la dirección de la gradiente que no se puede solucionar.

Por este motivo, se puede tener dos diferentes soluciones para este problema. La primera solución es el algoritmo de Horn-Schunck que determina la velocidad usando la ecuación de restricciones de la gradiente usando suavizamiento global. Este método emplea el campo de velocidad definido en términos de la función mínima determinada en la imagen [Thota et al., 2013]. Los autores Horn y Schunck agregan la restricción de un suavizado global para limitar la velocidad por medio de la minimización de la ecuación de la velocidad [Horn and Schunck, 1981].

A través de Horn-Schunck obtenemos una solución completa del flujo óptico, pero consume mucho tiempo computacional por las interacciones y la complejidad matemática que requiere. En consecuencia, se observa el algoritmo de Lucas-Kanade, que implementa el concepto del método *Least Square*. Dónde, se encuentra la velocidad que minimiza las restricciones de los errores [Lucas and Kanade, 1981].

El autor Cremers y sus colegas realizan la propuesta de estimar el flujo óptico con *Convolutional Neural Networks* (CNN) que incluye una capa que tiene vectores de características correlacionadas a las diferentes localizaciones de la imagen [Dosovitskiy et al., 2015]. Por otro lado, Weinzaepfel et. al. proponen un algoritmo de *matching* del descriptor adaptado al problema de flujo ópticos. Por lo que, los autores construyen el algoritmo como una arquitectura de múltiples etapas con 6 capas, convoluciones entrelazadas y *max-pooling* y una construcción *akin* para la CNN [Weinzaepfel et al., 2013].

Otra propuesta presenta un flujo óptico no lineal de orden reducida, en el cual se diseña una implementación simple y la generación de la estimación asintótica del rango de convergencia deseado [Keshavan and Humbert, 2017].

Por otro lado, el autor Pinto y sus colegas realizan el método llamado *Wise Optical Flow Clustering* (WOFC) y extracción del movimiento de los objetos por la operación de realizar dos operaciones consecutivas, evaluación y reinicio. Este método provee pistas confiables sobre el movimiento aparente; sin embargo, extraen regiones con características de movimiento similar en ese campo [Pinto et al., 2017].

El autor Revaud et. al. proponen una nueva estimación de flujo óptico para objetivos con desplazamientos largos con oclusión significativa (EpicFlow) [Revaud et al., 2015], este método consiste en dos pasos:

- *Matching* denso por interpolación de preservación de aristas de un conjunto *sparse* del *match*. La interpolación se basa en la elección apropiada de la distancia, es decir, una distancia geodésica consciente a la arista.
- Inicialización de la minimización de la energía variante con el *match* denso.

3.4 Estimación de Profundidad

Esta sección se centra en estimar la profundidad por medio de imágenes capturadas por la visión monocular, por lo que, se examina varios autores sobre este tema. El primer autor Longuet-Higgins estima el movimiento y la profundidad de una secuencia de imágenes monoculares sobre proyecciones de perspectiva [Fischler and Firschein, 1987]. De igual modo, Yen y sus colegas señalan que se debe optar por la proyección esférica donde una imagen plana es representada como una proyección central en unidades esféricas [Yen and Huang, 1983].

De manera similar, el Filtro de Kalman fue usado para obtener la profundidad teniendo en cuenta que la traslación debe ser paralela al plano de la imagen [Matthies et al., 1993]. Por lo que, se elabora un modelo cinemático general de estructuras de objetos tridimensionales. Los autores Broida y Chellappa han elaborado un sistema de estimación de profundidad empleando el filtro iterativo de Kalman Extendido [Broida and Chellappa, 1991].

Luego, se ha creado los algoritmos híbridos que se han presentado anteriormente, por lo que se realiza flujo óptico en los bordes del objeto para después fusionarlo con la etapa de estimación de filtro de Kalman. Esta etapa trabaja con la información de los píxeles vecinos [Zhuang et al., 1994], en cambio, el autor Hanmandlu y sus colegas usan proyecciones esféricas para obtener una ecuación simple. Esta ecuación es derivada para ser empleada en la estimación de profundidad por Filtro de Kalman [Hanmandlu et al., 1997].

Adicionalmente, se propone usar árboles de partición binaria, con sus siglas en inglés BPT, para obtener una jerarquía de regiones basada en las imágenes y estimar la profundidad por medio de las pistas de oclusión local [Palou and Salembier, 2012]. No obstante, el autor Mohamed et. al. proponen solucionar la recuperación de profundidad de un punto estático en tiempo real sin necesidad de la estimación iterativa [Hasan and Abdellatif, 2012].

De modo semejante, Lucas-Kanade proponen la estimación de profundidad para una secuencia de imágenes con pequeños desplazamientos basados en los vectores de flujo óptico realizando la estimación en una región de la imagen [Jose et al., 2014]. Por último, se usa el modelo de redes neuronales de convolución profunda para la estimación de la profundidad, que está basado en redes neuronales de convolución y el método de *superpixel pooling* [Liu et al., 2014].

Se puede concluir que la visión tiene diferentes técnicas existentes, que la mayoría de autores opta por el filtro de Kalman a pesar de la antigüedad del algoritmo. Además, el nuevo horizonte es fusionar KF con flujo óptico como Zhuang presenta [Zhuang et al., 1994]. También, el autor Isaac realiza una aproximación de profundidad utilizando un filtro gaussiano estándar, Kalman Extendido, por medio de una media y covarianza empírica [Issac et al., 2016].

Por otro lado, el autor Fayao Liu y sus colegas proponen un modelo que se basa en CNN y el método de *superpixel pooling* que incrementa la velocidad de procesamiento [Liu et al., 2016]. Además, el autor Mancini emplea *deep neural network* que es entrenada con imágenes reales y sintéticas, sacrificando un poco la precisión de la profundidad por una operación consistente, robusta y rápida [Mancini et al., 2016].

3.5 Cinemática Inversa

La cinemática inversa trata de encontrar los ángulos de cada articulación para que la herramienta del brazo manipulador llegue a la posición y la orientación deseada, por lo que, existen varios métodos. El primer método consta de emplear matrices de transformación homogénea para obtener la inversa de estas matrices. Siendo necesario que se conozca la posición final que se desea llegar para después solucionar las ecuaciones dadas por estas matrices.

Los autores Iliukhin y sus colegas presentan métodos geométricos basados en un modelo cinemáticos de un brazo manipulador de 5GDL [Iliukhin et al., 2017]. Por otro lado, Fei Liu et. al. realizan la solución inversa de las ecuaciones de cinemática basándose en el análisis de la configuración estructural [Liu et al., 2017].

El principal problema del primer método es el aumento de la cantidad de articulaciones por lo que aumenta las variables que se deben resolver, por esta razón se eleva la complejidad [Barrientos et al., 1997b]. Entonces, existe el método geométrico, que ubica las articulaciones en los ejes y trata de obtener geoméricamente la solución de cada variable. Sin embargo, tiene el mismo problema del método anterior cuando aumenta el número de articulaciones [Craig, 2006].

Para solucionar este problema se elaboró el desacoplo cinemático que trabaja con las primeras articulaciones por método geométrico y luego las otras articulaciones por medio de matrices de transformación homogénea [Craig, 2006].

También se aplican los algoritmos genéticos y redes neuronales, pero el principal problema es el tiempo y el cambio de entorno. Ya que, es diferente tener un brazo manipulador en una fábrica de automóviles que, en una fábrica textil, ya que el espacio de trabajo y funcionamiento son diferentes. Por lo que, se debe realizar previamente el entrenamiento de la red neuronal para tener una buena solución [Ali T. Hasan and Isa, 2011, Duka, 2014].

El autor Kölker propone la combinación de las características de una red neuronal Elman y técnicas evolutivas para obtener una mayor precisión en las soluciones. Este autor, aplica tres redes neuronales, cada una con entrenamiento separados para cada eje coordinado [KöKer, 2013].

Amouri y sus colegas proponen una meta-heurística que aplica optimización por partículas y algoritmos genéticos [Amouri et al., 2017]. Los autores Kucuk y Bingul realizan una nueva técnica de cinemática inversa que es obtenida por algoritmo de Newton-Raphson que resuelve las ecuaciones del modelo matemático del brazo manipulador [Kucuk and Bingul, 2014].

En resumen, se debe escoger la técnica de cinemática inversa tomando en cuenta el ambiente de trabajo dónde se va a emplear el brazo manipulador y el número de articulaciones, para observar la complejidad y obtener los ángulos de las articulaciones.

Capítulo 4

Propuesta

En los anteriores capítulos se revisa las diversas estrategias y fundamentos teóricos para realizar el sistema de control de un brazo manipulador. Por lo que, este capítulo se centrará en el desarrollo de la propuesta de esta investigación, en la que un brazo manipulador debe realizar el seguimiento de un objeto empleando un sistema de visión monocular como medio de percepción.

Esta propuesta surge por el problema de tener un mundo con movimientos tridimensionales, pero contando sólo con una percepción bidimensional. Por este motivo, se ha ideado aplicar la estrategia "divide y vencerás". Por lo que el problema se dividirá en dos sub-problemas que trabajaran en dos planos: el plano de la cámara y de la profundidad.

4.1 Estrategia "Divide y Vencerás"

La frase "divide y vencerás" es considerada una técnica de diseño de algoritmos. Sin embargo, suele ser considerada también una estrategia para resolver problemas empleada en diferentes campos de estudios. Esta estrategia se basa en los métodos que solucionan un problema de manera recursiva dividiéndolo en dos o más sub-problemas de igual tipo o similar. Se concluye que las soluciones de cada uno de los sub-problemas se combinan para dar la solución al problema original. Esta estrategia se ha empleado como base para los algoritmos eficientes para diferentes problemas. Por ejemplo, los algoritmos de ordenamiento *quicksort*, *mergesort*, entre otros; también para la multiplicación de números grandes, Katsuba, el análisis sintáctico como *top-down* y la transformada discreta de Fourier.

En este trabajo de investigación se emplea esta estrategia para dividir el problema principal en dos sub-problemas. Como se observa en la Figura 4.1, estos sub-problemas corresponden a los planos determinados de la siguiente forma:

- Plano de la cámara proporciona el movimiento que realiza el brazo manipulador en el plano XY . Por lo cual, se desea obtener la traslación y la rotación que realiza el objeto en un instante de tiempo.

- Plano de la profundidad será estimada, ya que sólo se tiene el plano de la cámara. Esta estimación se basa en los parámetros obtenidos del plano y la posición de la cámara para obtener el movimiento en el plano XZ .

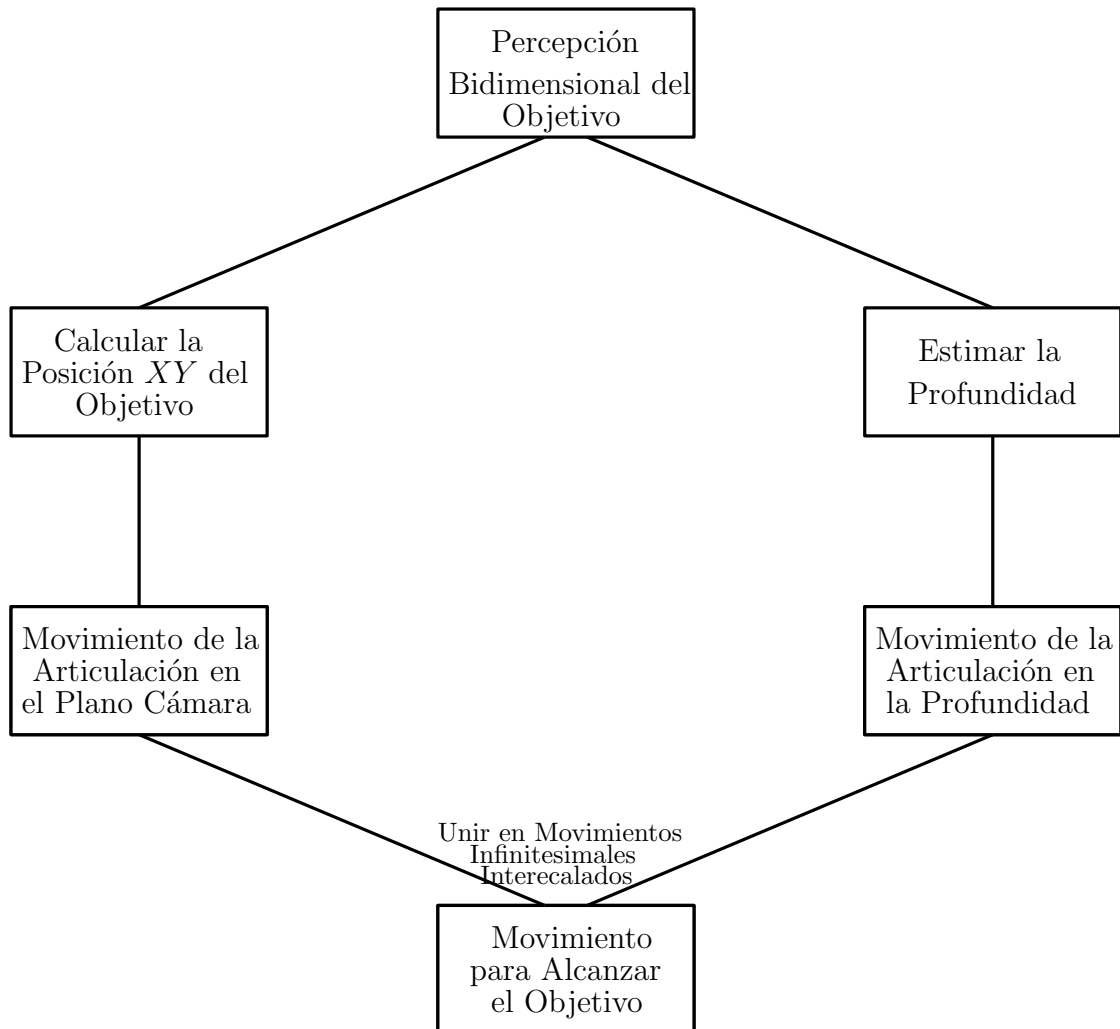


Figura 4.1: Diagrama de la implementación del sistema de control visual para el brazo manipulador del robot Komodo para alcanzar un objetivo.

4.1.1 Calcular la Posición XY del Objetivo

El primer sub-problema se da en el plano de la cámara, es decir, en el plano XY para obtener el movimiento del brazo manipulador. Para este plano se necesita dos componentes: la percepción del objetivo y el cálculo del movimiento que realiza el objetivo.

El primer componente es la percepción del objetivo por medio de la visión monocular que está localizado en la muñeca del brazo manipulador (Figura 4.2). En primer lugar, se define el espacio de color sobre el cual trabajar, considerando que será usado también para la estimación de profundidad.

Después, se debe escoger las características relevantes del objeto sin realizar mucho procesamiento computacional. El objeto seleccionado es una esfera, por lo que se ha optado

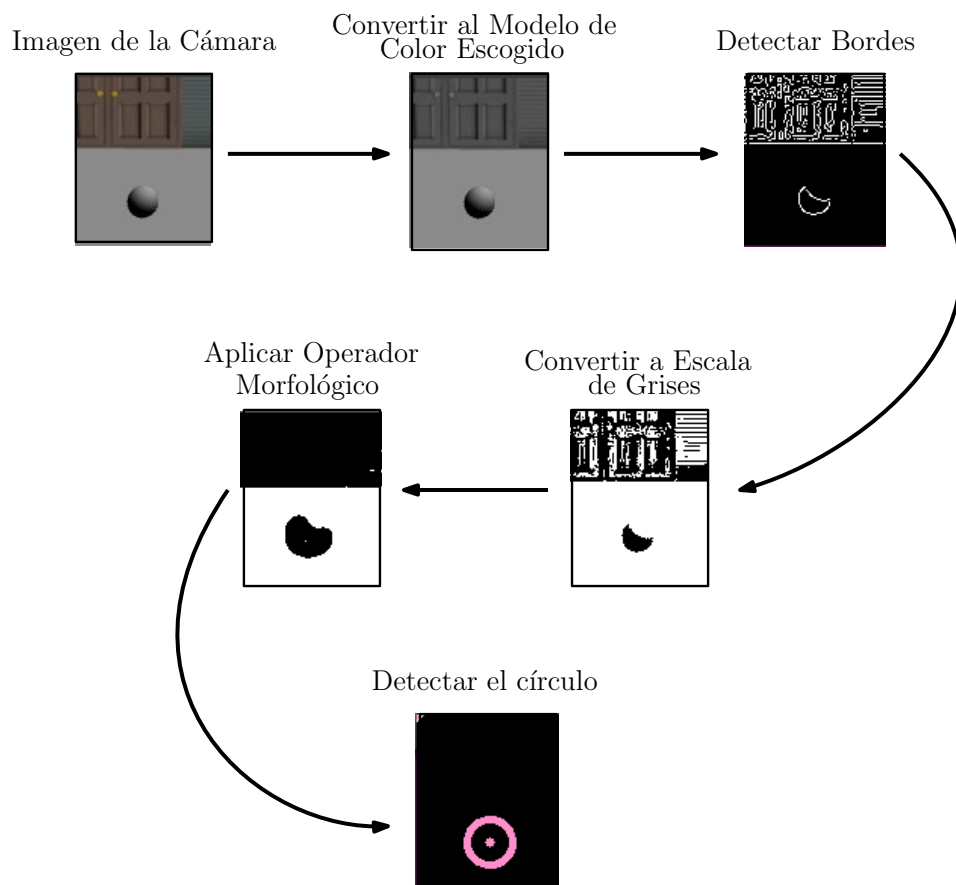


Figura 4.2: Partes del primer componente de percepción del objetivo.

por realizar la detección de bordes, por medio de la forma y el tamaño se detecta el objeto. Pero antes, se debe realizar la umbralización para obtener el objeto deseado y eliminar los no deseados. Además, se ha aplicado erosión a la imagen para suprimir el ruido restante.

Finalmente, al detectar el objeto deseado se obtiene el punto central por lo que se emplea el método de la transformada de Hough, que es usado para detectar objetos circulares con imperfecciones, es decir, círculos incompletos.

El segundo componente es obtener la translación que ha realizado el objeto, por el cual se ha empleado el flujo óptico de Lucas-Kanade para la región obtenida por medio de Hough. De este modo, se disminuye el procesamiento de la secuencia de imágenes. En primer lugar, se observa los puntos de la región que en este caso serán los puntos que están ubicados en la mitad de la esfera detectada. Después, se utiliza el flujo óptico entre la imagen en un tiempo $t - 1$ y la imagen en el tiempo t , para obtener el movimiento que ha realizado el objeto. Este movimiento es proporcional al movimiento del brazo manipulador en el plano XY .

Finalmente, se desea obtener la distancia que ha recorrido el objeto por lo que se ha realizado la diferencia del punto central del objeto detectado en el tiempo $t - 1$ y t . No obstante, se debe realizar una conversión de la distancia en píxeles hacia el desplazamiento en el mundo real.

4.1.2 Estimar la Profundidad

El segundo componente de la estrategia "divide y vencerás" es el plano de la profundidad. Al utilizar una cámara, no podemos obtener la profundidad, por esto se ha empleado la estimación. En este caso, se ha escogido la siguiente técnica descrita en la Figura 4.3 [Zhuang et al., 1994].

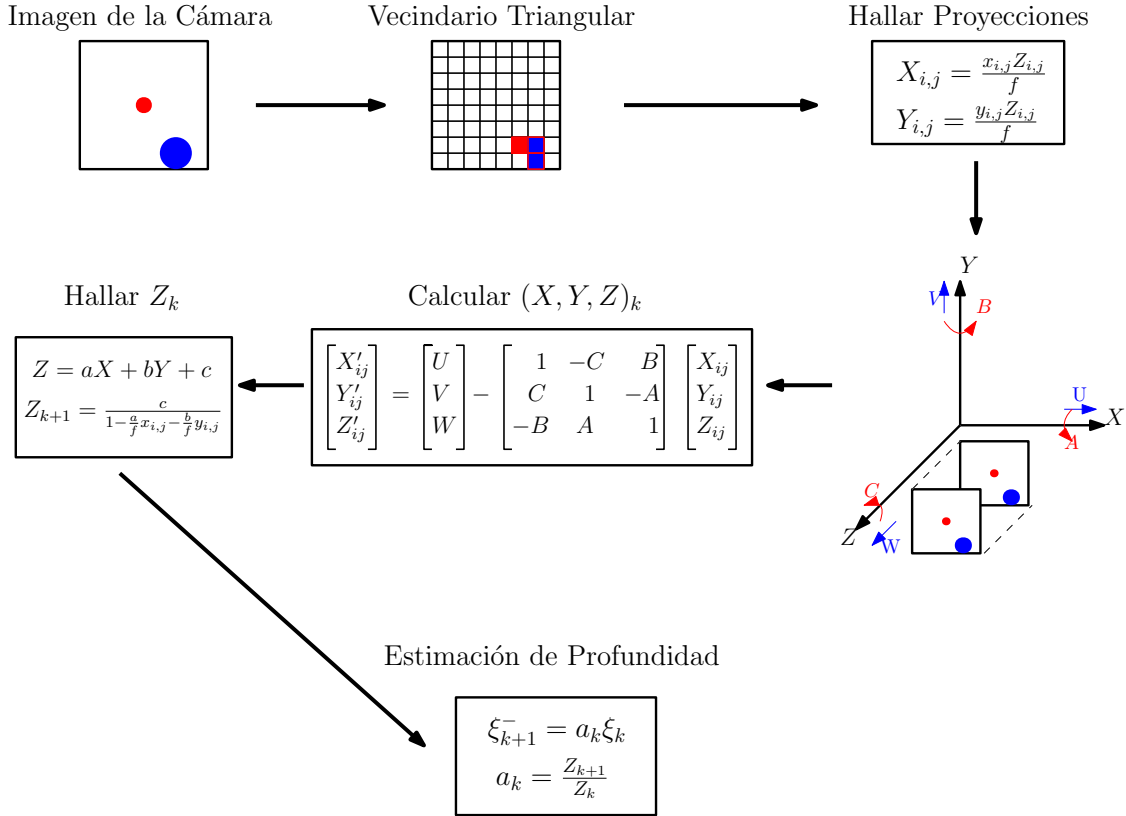


Figura 4.3: Partes de la Estimación de Profundidad (inspirado por [Zhuang et al., 1994]).

En primer lugar, se necesita de los siguientes parámetros para realizar esta técnica, la posición exacta que está localizada la cámara, el modelamiento del brazo manipulador y la obtención de la posición de cada articulación. Luego, se obtiene la posición central del objeto por medio de las técnicas utilizadas en el plano de la cámara. Por lo que, se debe hallar el vecindario triangular del punto central de la imagen ($I(i, j)$), que está compuesto por tres puntos $\{I(i, j), I(i, j + 1), I(i + 1, j + 1)\}$.

Luego, se realiza la proyección de cada uno de los puntos del vecindario, en el cual se aplican las siguientes ecuaciones 4.1 y 4.2 para cada punto:

$$X_{ij} = \frac{x_{ij}Z_{ij}}{f} \quad (4.1)$$

$$Y_{ij} = \frac{y_{ij}Z_{ij}}{f} \quad (4.2)$$

Estas dos ecuaciones dependen de la distancia focal f que está descrita por:

$$\frac{imageWidth}{2.0 * \tan(\frac{hfov}{2.0})} \quad (4.3)$$

Además, se debe encontrar la variación de la traslación y la rotación que están representadas por los parámetros U, V, W y A, B, C , que están descritas por las siguientes ecuaciones:

$$U = \frac{\Delta p_x}{\Delta t} \quad V = \frac{\Delta p_y}{\Delta t} \quad W = \frac{\Delta p_z}{\Delta t} \quad (4.4)$$

$$A = \frac{\Delta \alpha_x}{\Delta t} \quad B = \frac{\Delta \alpha_y}{\Delta t} \quad C = \frac{\Delta \alpha_z}{\Delta t} \quad (4.5)$$

La variación de la traslación es dada por la distancia obtenida del flujo óptico. En cambio, la rotación del objeto se halla por medio de la matriz de transformación homogénea del brazo manipulador, representada como:

$$\alpha_x = \arctan\left(\frac{O_z}{a_z}\right) \quad (4.6)$$

$$\alpha_y = \arctan\left(\frac{n_y}{n_x}\right) \quad (4.7)$$

$$\alpha_z = \arctan\left(-\frac{n_z}{O_z} \sin \alpha_x\right) \quad (4.8)$$

Seguidamente, necesitamos obtener el plano compuesto por tres puntos (R_1, R_2, R_3). Este paso, nos ayudará para obtener la estimación de la profundidad. Se debe tener en claro la ecuación del plano $Z = aX + bY + c$, donde los puntos son descritos como:

$$R_1 = (X_{i,j}, Y_{i,j}, Z_{i,j})_{k+1} \quad (4.9)$$

$$R_2 = (X_{i,j+1}, Y_{i,j+1}, Z_{i,j+1})_{k+1} \quad (4.10)$$

$$R_3 = (X_{i+1,j+1}, Y_{i+1,j+1}, Z_{i+1,j+1})_{k+1} \quad (4.11)$$

$$(4.12)$$

Luego, se halla el punto de intersección con el plano para obtener la estimación de la profundidad del *frame* actual por medio de las variables a, b, c :

$$Z = aX + bY + c \quad (4.13)$$

$$Z_{i,j} = \frac{a}{f}x_{i,j}Z_{i,j} + \frac{b}{f}y_{i,j}Z_{i,j} + c \quad (4.14)$$

$$Z_{i,j} = Z_{i,j}\left(\frac{a}{f}x_{i,j} + \frac{b}{f}y_{i,j} + \frac{c}{Z'_{i,j}}\right) \quad (4.15)$$

$$1 = \frac{a}{f}x_{i,j} + \frac{b}{f}y_{i,j} + \frac{c}{Z'_{i,j}} \quad (4.16)$$

$$Z'_{i,j} = \frac{c}{1 - \frac{a}{f}x_{i,j} - \frac{b}{f}y_{i,j}} \quad (4.17)$$

Por último, se ha empleado el Filtro de Kalman para actualizar el error de la estimación de la profundidad, y así, perfeccionarla. Se debe tener en cuenta que el KF es un conjunto de ecuaciones matemáticas que implementan un estimador de tipo *predictor-corrector*, por el cual se minimiza la covarianza del error. Por ende, este filtro se basa en el estado de estimación a *priori* (ξ_{k+1}) y la covarianza del error asociado a la estimación a *priori* (ρ_{k+1}^-).

$$\xi_{k+1}^- = a_k \xi_k \quad (4.18)$$

$$\rho_{k+1}^- = a_k^2 \rho_k + q_k \quad (4.19)$$

$$\xi_{k+1} = \xi_{k+1}^- + K_{k+1} [\eta_{k+1} - H_{k+1} \xi_{k+1}^-] \quad (4.20)$$

$$P_{k+1} = [I - K_{k+1} H_{k+1}] P_{k+1}^- \quad (4.21)$$

$$K_{k+1} = P_{k+1} H_{k+1}^T R_{k+1}^{-1} \quad (4.22)$$

En este caso, se va a realizar el error estimado como la diferencia entre la estimación a *priori* con la estimación a *posteriori*.

$$a_k = \frac{Z_{k+1}}{Z_k} \quad (4.23)$$

$$\epsilon_{k+1}^- = a_k \epsilon_k \quad (4.24)$$

4.1.3 Movimiento de la Articulación

El movimiento de la articulación se realiza por medio del método de cinemática inversa. Por lo que, se ha implementado el controlador PID que permite realizar el sistema de control por medio de la retroalimentación del error entre el valor obtenido y el valor deseado.

Este controlador consiste en tres componentes: proporcional, integral y derivativo. En este caso, sólo se emplea el componente proporcional para cada uno de los ejes que se realizará el movimiento.

Al calcular la posición que se ha movido el objeto en los ejes X y Y , se convierte este desplazamiento de píxeles a centímetros y se aplica este tipo de controlador para hacer el movimiento en estos dos ejes.

Luego, se realiza el movimiento en los ejes Y y Z con el mismo controlador PID. Finalmente, se realiza de forma intercalada infinitesimalmente del movimiento de XY y YZ para realizar el movimiento tridimensional.

Capítulo 5

Resultados

En este capítulo se va a sub-dividir en tres partes fundamentales, en primer lugar, la descripción completa del hardware utilizado y variables de implementación. Luego, se observará los resultados globales del movimiento que realiza el brazo manipulador en las diferentes pruebas. En tercer lugar, se va a mostrar las pruebas de las etapas fundamentales para la realización de esta investigación.

5.1 Parámetros de Implementación

5.1.1 Hardware

Esta propuesta a utilizado el brazo manipulador del robot Komodo de la compañía *Robotican* que se muestra en la Figura 5.1.

El modulo del brazo (Fig. 5.2) consta de los siguientes elementos:

- Brazo de 7 GDL.
- Posición, velocidad y retroalimentación de corriente para todas las articulaciones.
- Carga útil: $0.4Kg$.
- Cámara ubicada en la muñeca del brazo.
- Rango: $80cm$.
- Puede ser usado como un brazo manipulador *standalone*.
- Peso: 4.5 Kg (con batería).
- Batería: *LiFePO4* 12V 6.4Ah.
- MTBC: 2 horas.

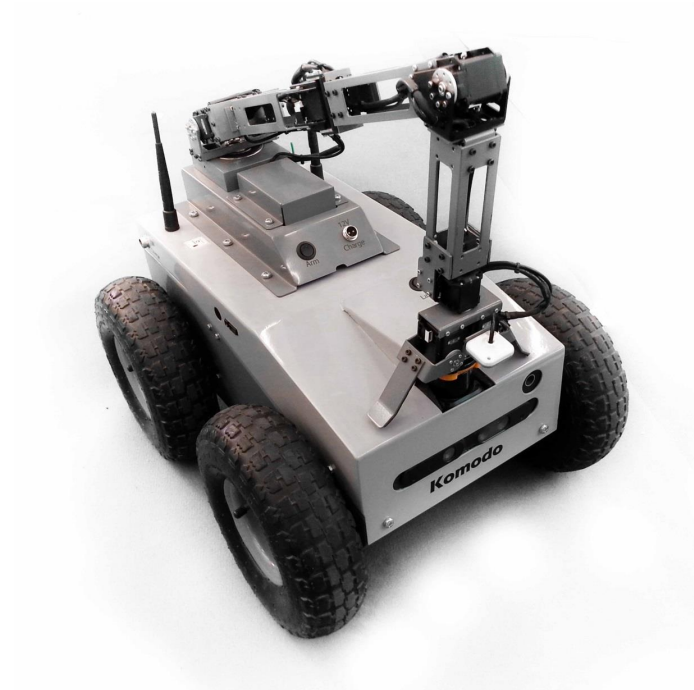


Figura 5.1: Robot Komodo [RoboTiCan, 2013].



Figura 5.2: Modulo del Brazo del Robot Komodo (fotografía del brazo manipulador del Robot Komodo).

5.1.2 Software

La principal herramienta que usamos es el *framework Robot Operating System (ROS)*. Este *framework* es un conjunto de herramientas y convenciones que permite simplificar las tareas

para crear robots complejos y robustos a través de una red variada de plataformas robóticas [O’Kane, 2013].

ROS se basa en un número de componentes, así como un conjunto de *drivers* que permite leer los datos de los sensores y el envío de comandos a los motores. Adicionalmente, proporciona una infraestructura que permite mover los datos y conectar diversos componentes al sistema de control del robot e incorporar propios métodos en diversos lenguajes de programación [Quigley et al., 2015].

Se ha simulado el brazo manipulador presentado en la anterior sección en Gazebo, que nos ofrece un ambiente estable para realizar pruebas en una variedad y sistemas de múltiples robots con los diferentes sensores de forma realista. Los objetos que se emplean mantienen correlación con las propiedades físicas como la masa, la velocidad y la fricción de sus contrapartes reales [Cañas et al., 2014].

Asimismo, proporciona diferentes ambientes y control de la iluminación del espacio de trabajo. Por lo que, existe tres tipos de ambientes como:

- Ambientes simples, se enfoca en escenarios para realizar tareas en un espacio de trabajo específico. Por ejemplo, un brazo manipulador de una fábrica que recoge objetos de una banda transportadora en un espacio fijo.
- Ambientes interiores, el escenario simula una estación de trabajo, es decir se puede realizar diferentes tareas como el planeamiento de caminos para que un brazo manipulador móvil llegue a su destino en un almacén con objetos móviles.
- Ambientes exteriores, es decir, un entorno no estructurados dónde puede influir la naturaleza. Es para realizar pruebas en robots aéreos, robots acuáticos, robots de rescate, etc.

5.1.3 Variables de implementación

Se ha debido realizar el modelamiento matemático del brazo manipulador teniendo en cuenta la cámara (Fig. 5.3), por lo que se ha utilizado la notación de D-H que esta mostrado en la Tabla 5.1.

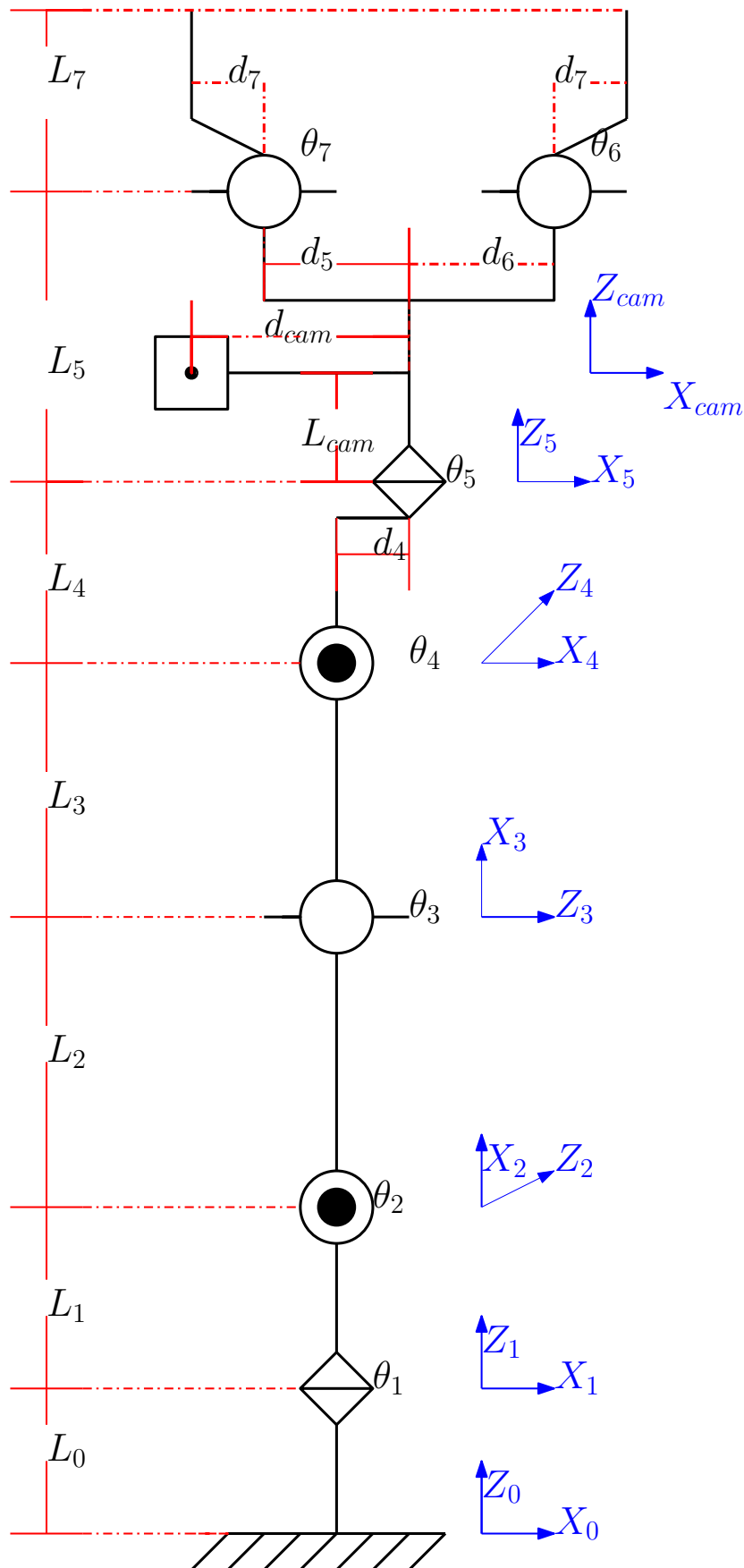


Figura 5.3: Notación de D-H basado en la cámara del brazo manipulador del robot Komodo.

Articulación	a (cm)	α (radian)	d (cm)	θ (radian)
1	0.0	0.0	13.8	θ_1
2	0.0	$-\frac{\pi}{2}$	0.0	$-\frac{\pi}{2} + \theta_2$
3	27.0	$-\frac{\pi}{2}$	0.0	$-\theta_3$
4	22.3	$-\frac{\pi}{2}$	0.0	$-\frac{\pi}{2} - \theta_4$
5	1.0	$-\frac{\pi}{2}$	24.0	θ_5
cámara	-6.5	0.0	4.5	0.0

Tabla 5.1: Parámetros de D-H de la base a la cámara del brazo manipulador del Komodo.

Además, se tiene la siguiente inicialización de las variables. Las primeras variables son parámetros intrínsecos de la cámara:

- $image_width = 640$
- $hfov = 2.0$
- $focal_length = image_width / (2.0 * \tan(hfov/2.0))$
- $centerCamX = 640.0/2.0$
- $centerCamY = 480.0/2.0$

Por último, las variables obtenidas por las pruebas que se han realizados:

- $threshold_value = 78$
- $max_binary_value = 255$
- $countCalibration = 0$
- $updateRate = 33.3333$
- $alphaPID = 0.0349066$

5.2 Pruebas Globales

Las pruebas que se han llevado a cabo con una esfera de $6cm$ de diámetro. Además, se ha realizado pruebas en tres diferentes casos de estudios: la profundidad, la altura y la distancia al centro de la cámara. Se ha realizado un estudio de 20 pruebas cada una elaborada en 1 segundo. Por lo que, se ha establecido tres tipos de iluminación:

- Iluminación constante
- Iluminación enfocada a la cámara

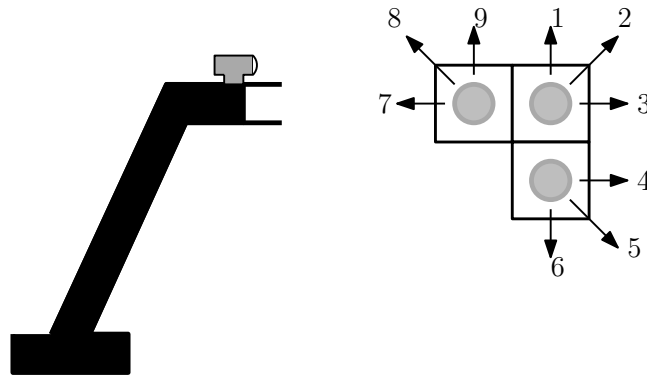


Figura 5.4: Casos de prueba según la iluminación y la posición.

- Iluminación enfocada a la esfera

En la Figura 5.4 representa los casos de pruebas que se han realizado, dónde los números 1, 4, 7 representan la iluminación normal. Las pruebas 2, 5, 8 se centra la iluminación en la cámara. Finalmente, las pruebas se realizaron con una iluminación centrada en la esfera mostrada en 3, 6, 9. Además, los casos 1, 2 y 3 representan la profundidad de 60 cm, 52 cm de altura y 40 cm al lado derecho del eje central de la cámara. A parte de ellos, los casos 4, 5 y 6 tienen las mismas medidas sólo que con una menor altura (10 cm). Los otros casos, se realizan a una profundidad de 30 cm, una altura de 52 cm y 40 cm al lado derecho del eje central de la cámara.

En la Tabla 5.2 se muestran los datos de prueba en el cual se indican el promedio y la desviación estándar de los casos de estudios dados en centímetros. Además, se puede observar una desviación mínima en el caso 4 que consta de una iluminación normal.

Tabla 5.2: Promedio y Desviación estándar de los casos de estudio en cm.

	μ_x	σ_x	μ_y	σ_y	μ_z	σ_z
Caso 1	12.11	6.05	2.79	1.72	10.21	6.05
Caso 2	11.20	6.50	9.29	7.12	17.15	7.05
Caso 3	11.13	4.03	2.08	1.69	7.53	5.44
Caso 4	5.05	2.21	4.80	2.77	3.00	1.45
Caso 5	4.02	1.20	7.16	3.80	6.08	4.60
Caso 6	4.02	1.20	7.17	3.80	8.08	4.60
Caso 7	6.36	3.19	10.04	5.58	8.34	6.79
Caso 8	5.68	4.36	9.78	4.49	8.83	3.80
Caso 9	6.84	3.04	10.15	5.34	6.73	6.40

El primer caso de estudio realiza una profundidad de 60cm con una altura de 52 cm, dada a una distancia de 40cm del centro de la cámara al lado derecho. En este estudio, se realiza una desviación estándar de 8.9cm por medio de una trayectoria suavizada que se aprecia en la Figura 5.5 con respecto al eje X . En síntesis, se tiene una mayor incertidumbre con la esfera al costado. Sin embargo, en el caso del eje Y se analiza la desviación estándar de 1.7 cm; en el cual se observa oscilaciones en la Fig. 5.6, ya que este eje recibe dos controladores PID por los movimientos en el plano de la cámara y el de la profundidad.

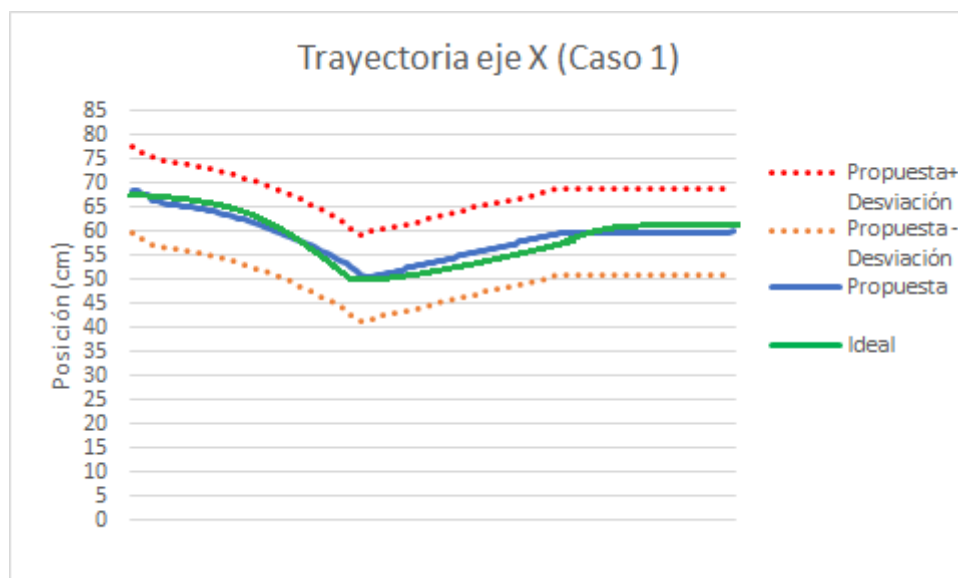


Figura 5.5: Trayectoria del eje X del caso 1.

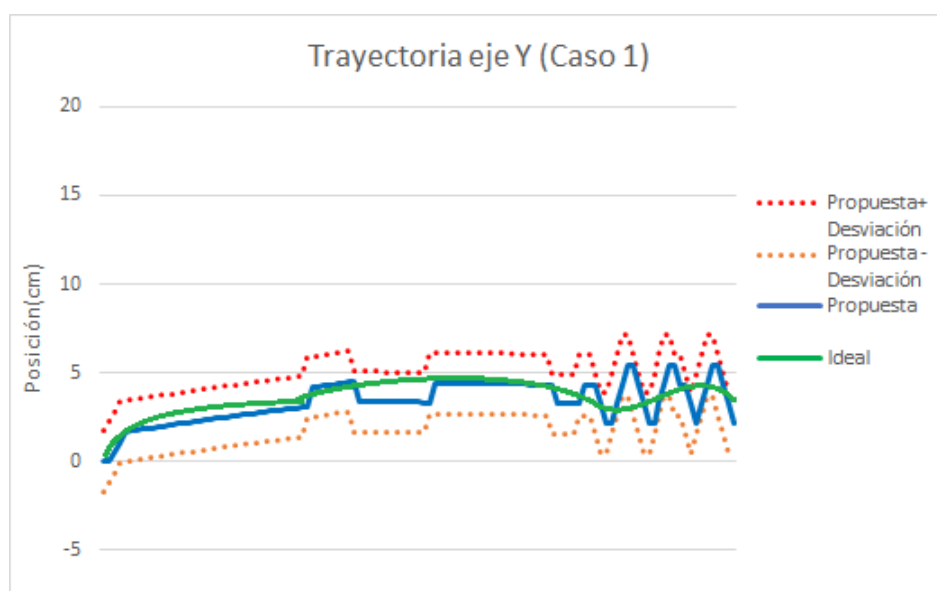


Figura 5.6: Trayectoria del eje Y del caso 1.

En otro aspecto, en el eje Z se observa 6 cm de desviación estándar, resulta que la estimación de profundidad depende del plano XY incrementando la incertidumbre del eje X . Por ende, se eleva el error en la Figura 5.7.

El segundo caso de estudio se muestra la desviación estándar de 9.0 cm con una trayectoria suavizada, que se observa en la Figura 5.8 en el eje X . Se muestra la esfera al costado del eje central de la cámara por lo que se da una mayor incertidumbre. Sin embargo, las pruebas realizadas en el eje Y se tiene una desviación estándar de 7.1 cm. A causa de que el objeto sale del rango de visión de la cámara en algunos instantes de tiempo, debido a la iluminación que permite detectar una región pequeña de la esfera (Fig. 5.9).

Además, en el eje Y como se menciona en el caso 1, se examinan oscilaciones por que se ha tenido realizar dos controladores para el mismo eje, lo cual ha incrementado las oscilaciones

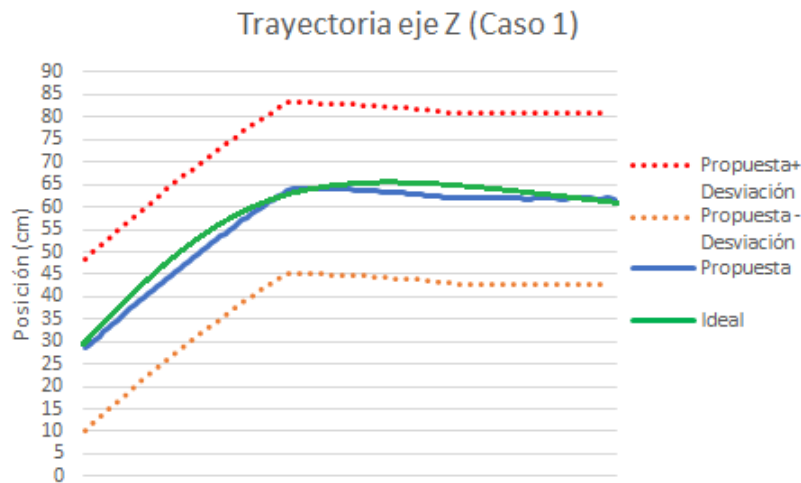


Figura 5.7: Trayectoria del eje Z del caso 1.

que se presentan en la Figura 5.9.

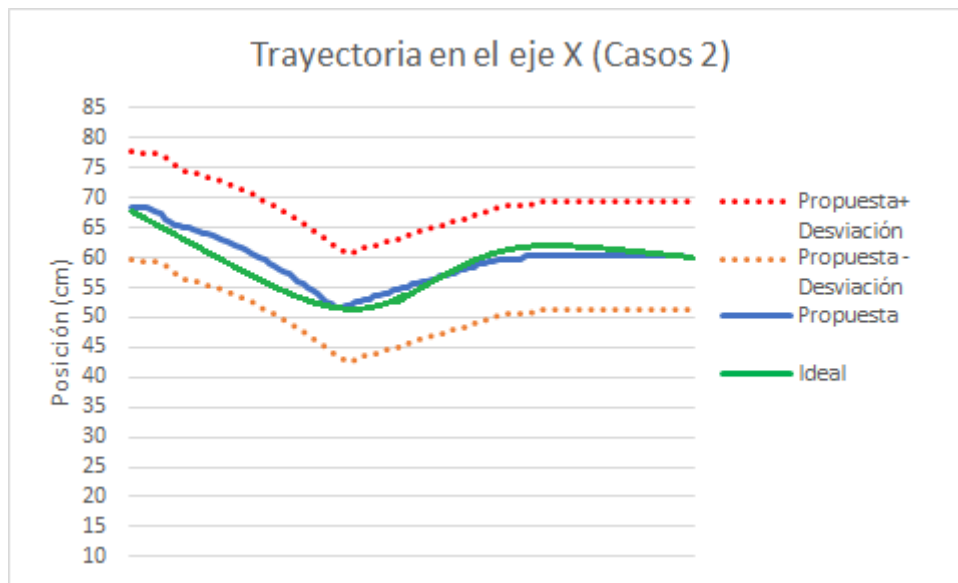


Figura 5.8: Trayectoria del eje X del caso 2.

Sin embargo, el eje Z se tiene la desviación estándar es de 5.8 cm, ya que la estimación de profundidad depende del plano XY por lo cual la incertidumbre es mayor en el eje X (Figura 5.10).

El tercer caso de estudio se muestra la desviación estándar de 5.0 cm a través de una trayectoria suavizada que se aprecia en la Figura 5.11 con respecto al eje X . Por tener el mismo problema que se ha visto anteriormente, se observa que en el eje Y con una desviación estándar de 1.7 cm. Este error es pequeño por la altura de la cámara, disminuyendo la incerteza (Fig. 5.12), además se observa oscilaciones que se ha explicado en los casos anteriores que es por realizar dos controladores a un mismo eje.

En otro aspecto, el eje Z tiene una desviación estándar de 6.4cm por medio de la estimación de profundidad que depende del plano XY , determinado por la Figura 5.13.

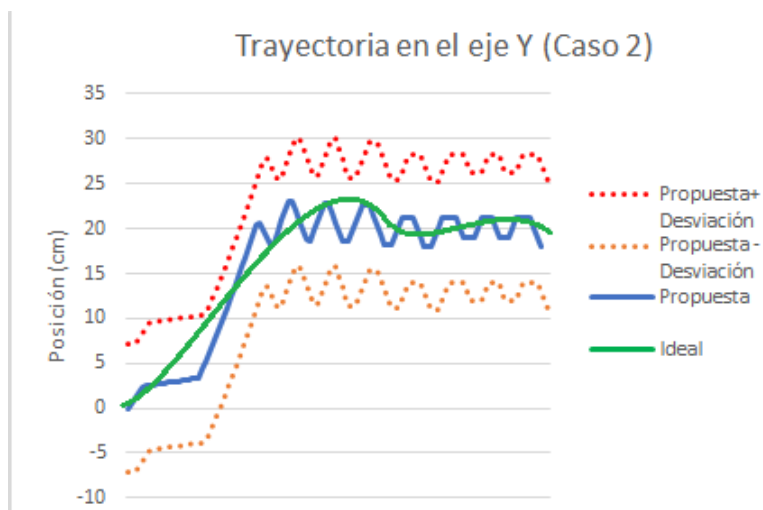


Figura 5.9: Trayectoria del eje Y del caso 2.

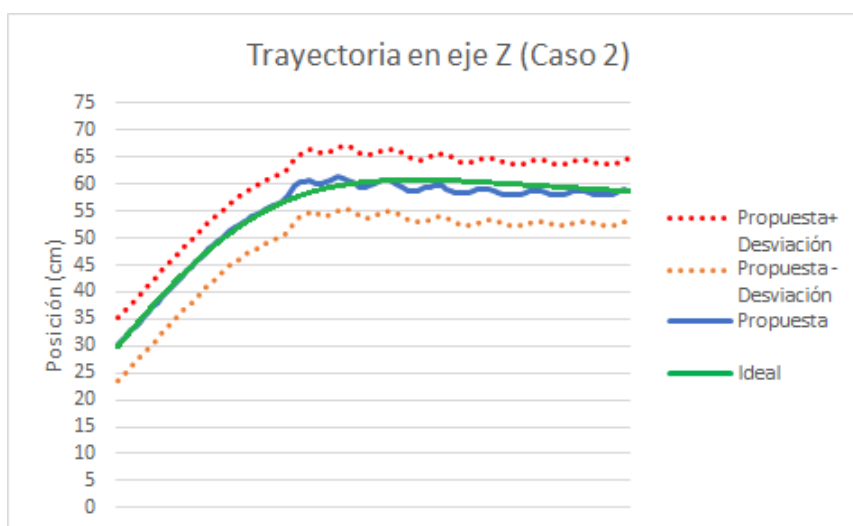


Figura 5.10: Trayectoria del eje Z del caso 2.

5.3 Pruebas Específicas

Las pruebas específicas que se han realizado es para dos componentes fundamentales. El primer componente es con respecto a la percepción bidimensional del objeto y la detección del objeto. El segundo componente son las matrices de transformación homogénea, es decir, la cinemática directa. Ya que, para tener una solución precisa de la estimación de profundidad se debe observar que el modelado es correcto.

5.3.1 Pruebas de Percepción

En el capítulo de la propuesta se ha descrito la percepción bidimensional de la imagen de la cámara, se debe tener en cuenta que el objeto tiene un diámetro de 3 cm de color gris. Por lo que, se ha tomado 100 imágenes consecutivas con la cámara del brazo manipulador en diferentes tipos de fondo (claro, oscuro y textura) que se perciben en la Tabla 5.4. Además, se ha probado

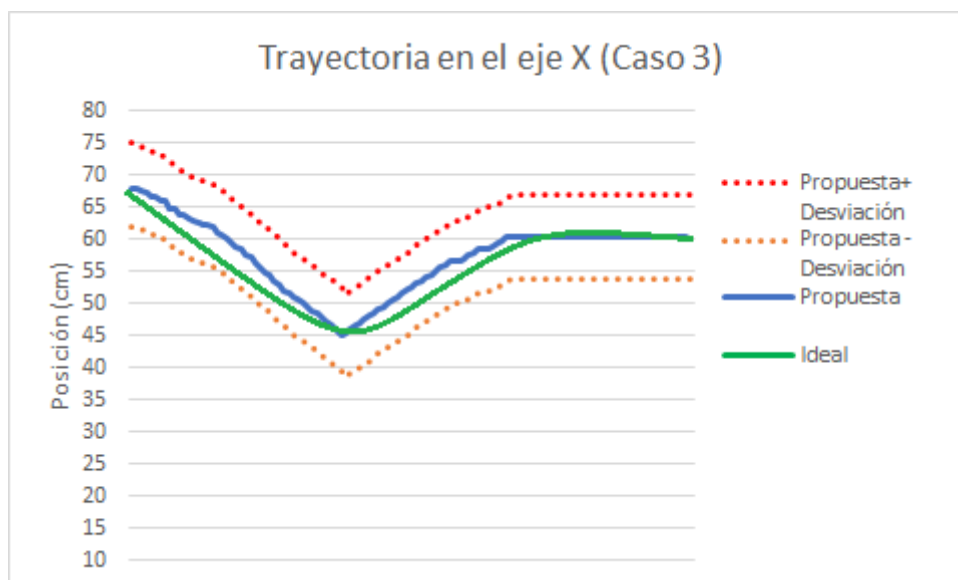


Figura 5.11: Trayectoria del eje X del caso 3.

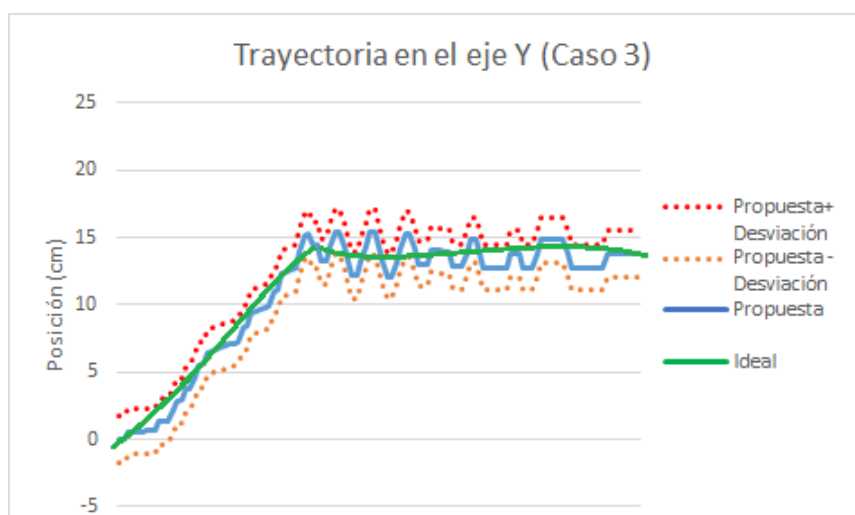
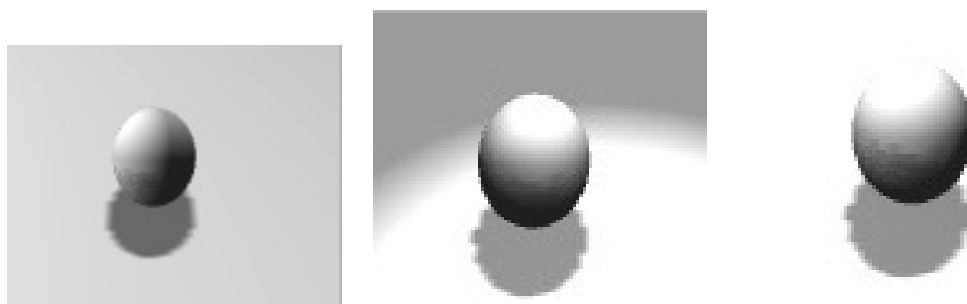


Figura 5.12: Trayectoria del eje Y del caso 3.

en diferentes tipos de iluminación que se muestran en la Tabla 5.3.

Se han elaborado las curvas de ROC, que se basa en los valores Verdaderos Positivos (VPR) y Falsos Positivos (FPR). En otros términos, los casos en que se ha detectado la esfera en la imagen, y en caso contrario que se haya detectado otro objeto como la esfera (Figura 5.14).



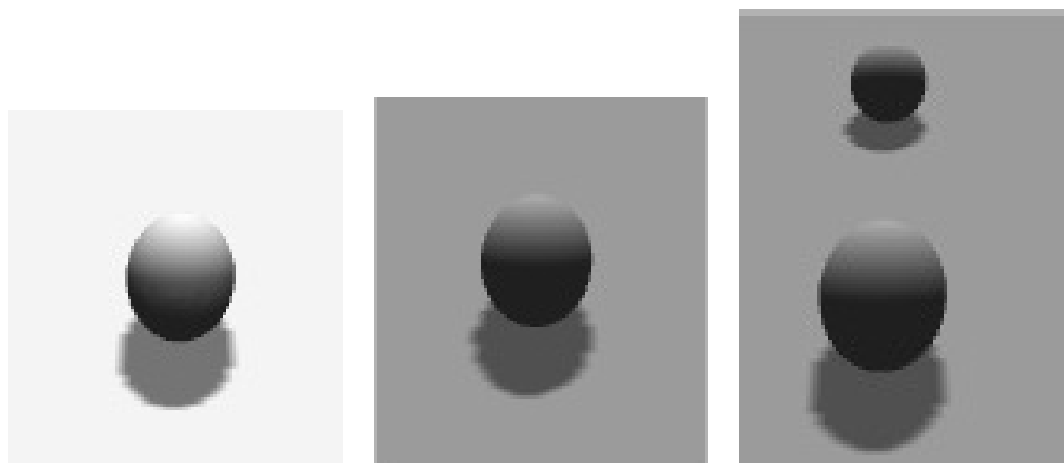


Tabla 5.3: Diferentes perspectivas de iluminación de la esfera.

Esta prueba se tiene un resultado óptimo en las imágenes, a pesar de los cambios de iluminación a través de una sensibilidad de 0.65 y un radio falso positivo de 0.25. Este trabajo se ha realizado en el espacio de color *Hue, Saturation and Lightness* (HSL) en el canal de luminancia, se tiene un error alto por la iluminación oscura.

Sin embargo, después se analizó las pruebas con diferentes tipos de fondo, lo cual mejora la sensibilidad y disminuye el radio de falsos positivos. Por ejemplo, en un fondo oscuro se analiza un índice VPR de 0.95 con FPR de 0.05. En cambio, en un fondo claro se obtiene un radio de verdaderos positivos de 0.85 decayendo en un 10%, esto fue ocasionado porque algunas imágenes de prueba no pueden ser umbralizadas por que el objeto no era detectado. Pero, por lo menos se ha mejorado en un 15% la sensibilidad en las pruebas con objetos con fondo.

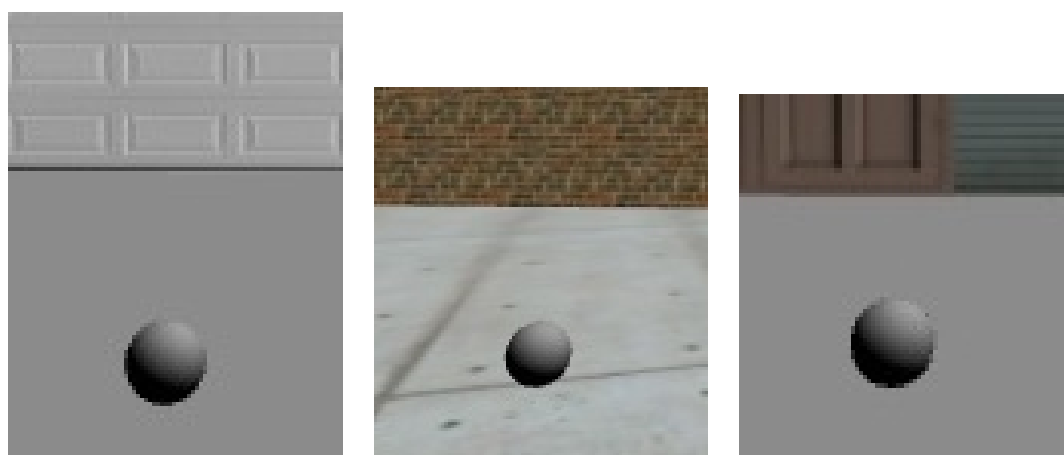




Tabla 5.4: Casos de pruebas de la etapa de percepción en diferentes fondos.

Por último, se realizó pruebas con fondo en diferentes texturas, mostrando un resultado en VPR de 0.97 y FPR de 0.04. Esto, se debe que el fondo se tiene un color homogéneo por lo cual no se ha tenido problemas para reconocer el objeto.

5.3.2 Cinemática Directa

Se debe realizar pruebas exhaustivas en el modelado del brazo robótico, ya que todos los demás módulos dependen del modelado de la cadena cinemática del brazo manipulador. Sin embargo, para probar el modelado se realiza por medio del módulo de cinemática directa que permite obtener la posición final de la herramienta por medio de los ángulos de cada articulación del brazo robótico.

En este caso, se ha realizado pruebas con el modelado del brazo manipulador con respecto a la cámara que se muestra en el capítulo anterior. Por lo cual, se ha realizado veinte pruebas para cada uno de las articulaciones para un movimiento de 0.174 radianes en el rango de $-\frac{\pi}{2}$ al $\frac{\pi}{2}$ radianes, dónde se ha calculado el error promedio.

La primera articulación que se analiza es *base_rotation*, dónde se mide el error en los tres ejes coordenados. Por lo que, se muestra un error de 0.0005cm con respecto a la cámara en el eje *X* (Tabla 5.5). Este error es imperceptible, por lo cual permite validar los ejes coordenados del modelo del brazo manipulador en esta articulación.

La segunda articulación se analiza el *shoulder* que proporciona el movimiento en profundidad, por lo que esta articulación se tiene un margen de error de 0.0091 cm en el eje *Z* con respecto a la cámara. Esta articulación está formada por dos accionadores, ya que el peso de todo el brazo manipulador es soportado por esta articulación. Además, el peso propio de las otras articulaciones sobrecarga el motor lo que impulsa que este sea desplazado unos centímetros hacia abajo. Por este motivo, la articulación tiene mayor error promedio, como se puede apreciar en la Tabla 5.6.

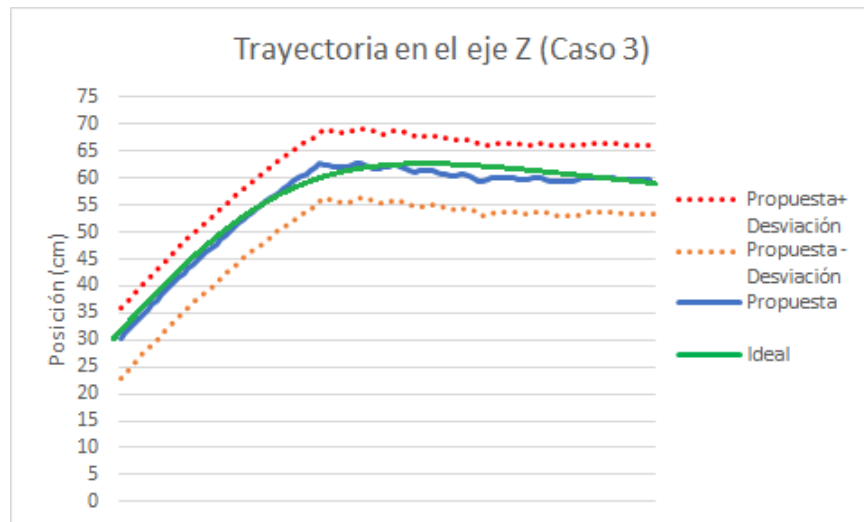


Figura 5.13: Trayectoria del eje Z del caso 3.

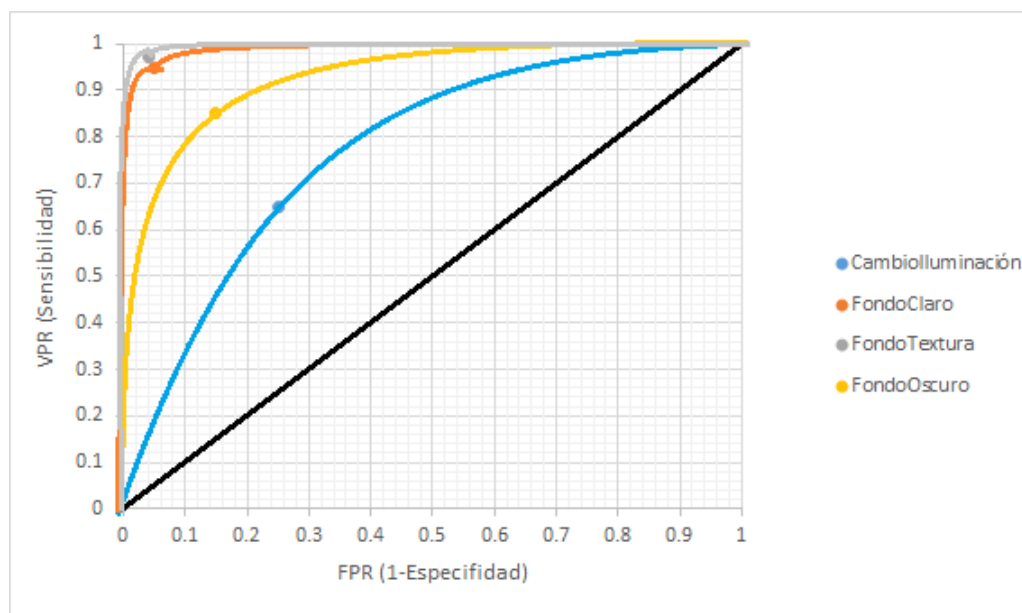


Figura 5.14: Curva de ROC de la percepción bidimensional del objeto en los diferentes casos de prueba.

Posición	Error X	Error Y	Error Z
Cámara	0.0005	0.0001	0.0000
<i>Wrist</i>	0.0001	0.0000	0.0000
<i>Elbow2</i>	0.0000	0.0000	0.0000
<i>Elbow1</i>	0.0000	0.0000	0.0000
<i>Shoulder</i>	0.0000	0.0000	0.0000
<i>Base_Rotation</i>	0.0000	0.0000	0.0000

Tabla 5.5: Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en *base_rotation*.

De manera semejante, la articulación *elbow1* permite el movimiento en el eje X y la cámara es perpendicular al eje de movimiento de esta articulación. Por este motivo el error que

Posición	Error X	Error Y	Error Z
Cámara	0.0020	0.0000	0.0091
<i>Wrist</i>	0.0015	0.0000	0.0086
<i>Elbow2</i>	0.0010	0.0000	0.0058
<i>Elbow1</i>	0.0006	0.0000	0.0032
<i>Shoulder</i>	0.0000	0.0000	0.0000
<i>Base_Rotation</i>	0.0000	0.0000	0.0000

Tabla 5.6: Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en *shoulder*.

presenta es 0.0044 cm en el eje Z con respecto a la cámara. De igual forma que la anterior articulación, el peso de las articulaciones siguientes se observa una pequeña disminución de la posición real (Tabla 5.7).

Posición	Error X	Error Y	Error Z
Cámara	0.0000	0.0011	0.0044
<i>Wrist</i>	0.0000	0.0010	0.0040
<i>Elbow2</i>	0.0000	0.0005	0.0019
<i>Elbow1</i>	0.0000	0.0000	0.0000
<i>Shoulder</i>	0.0000	0.0000	0.0000
<i>Base_Rotation</i>	0.0000	0.0000	0.0000

Tabla 5.7: Error Promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en *elbow1*.

Seguidamente, se analizará la articulación *Elbow2*, en el cual se obtuvo un error máximo de 0.0021 cm con respecto a la cámara en el eje Z (Tabla 5.8). Se ha obtenido este error por la desviación presente entre el eje del motor con la anterior articulación, por consiguiente, se ha dificultado la búsqueda de esta imprecisión observada después de varios intentos de modelado se identificó esta pequeña desviación en la estructura propia del brazo manipulador.

Posición	Error X	Error Y	Error Z
Cámara	0.0007	0.0000	0.0021
<i>Wrist</i>	0.0005	0.0000	0.0018
<i>Elbow2</i>	0.0000	0.0000	0.0000
<i>Elbow1</i>	0.0000	0.0000	0.0000
<i>Shoulder</i>	0.0000	0.0000	0.0000
<i>Base_Rotation</i>	0.0000	0.0000	0.0000

Tabla 5.8: Error promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en *elbow2*.

Por último, las pruebas mostradas en la articulación *wrist* en la Tabla 5.9, se contempla que el error con respecto a la cámara en el eje Z . El error máximo obtenido es de 0.0021 cm, ya que en la anterior articulación existe pequeñas variaciones en la estructura del brazo manipulador, especialmente en los ejes de movimiento, por lo que no se ha logrado medir correctamente.

Se puede concluir que las pruebas exhaustivas de la cinemática directa que validan el modelo de la cadena cinemática del brazo manipulador, ya que los errores presentados son muy pequeños, casi imperceptibles, por lo que no afecta al sistema de control. Además, se puede analizar que los errores más altos se dan con respecto a la cámara y en el eje Z ya que la desviación de la estructura propia del brazo manipulador que se ha visto en el modelo dado en el capítulo de la implementación.

Posición	Error X	Error Y	Error Z
Cámara	0.2601	0.0002	0.0000
<i>Wrist</i>	0.0000	0.0000	0.0000
<i>Elbow2</i>	0.0000	0.0000	0.0000
<i>Elbow1</i>	0.0000	0.0000	0.0000
<i>Shoulder</i>	0.0000	0.0000	0.0000
<i>Base_Rotation</i>	0.0000	0.0000	0.0000

Tabla 5.9: Error promedio en centímetros de los tres ejes coordenados en la articulación de las pruebas en *wrist*.

Capítulo 6

Conclusiones y Trabajos Futuros

Este capítulo se dividió en cuatro partes, la primera parte se observa las conclusiones del trabajo realizado. Luego, se analiza los problemas encontrados en la implementación de la propuesta de esta investigación. Seguidamente, se observa las limitaciones y recomendaciones de la propuesta que se ha mencionado con anterioridad. Por último, se analiza los posibles trabajos futuros para esta línea de investigación.

6.1 Conclusiones

En las pruebas globales se observa la curva de estabilidad del error de estimación del plano profundidad que depende del plano XY , por lo cual se puede observar que en la tabla del promedio y desviación estándar de los casos de prueba que se han realizado. En las figuras mostradas que en el eje Y presenta oscilaciones, esto se debe a que aplicamos dos veces el controlador **PID!** (**PID!**), ya que se realiza movimiento en el plano de la cámara y el plano de la profundidad. A pesar de las oscilaciones mostradas, se concluye que se presenta un movimiento parecido al ideal con pequeños errores.

Tomando en cuenta, las conclusiones presentadas, podemos observar que el brazo manipulador tiene un movimiento suave y limpio, ya que el controlador realiza una retroalimentación visual en cada instante. Por lo que, se muestra en los resultados la trayectoria suavizada en los diferentes casos de estudio que se han presentado en el anterior capítulo que muestran la trayectoria ideal contra la trayectoria de la propuesta.

Para desarrollar los algoritmos de control se tiene una dependencia del modelo cinemático, por lo que al tener un modelado erróneo nos lleva a un resultado insatisfactorio. Por este motivo, se ha probado el modelado cinemático del brazo, obteniendo un error menor a 0.05 cm, por lo que se concluye que el modelado es correcto (publicación en el Simposio Latinoamericano de Robótica [Lovon-Ramos et al., 2016]).

Además, se han realizado las pruebas específicas del módulo de percepción del objeto por medio de las curvas de *Receiver Operating Characteristic*. En estas pruebas, se observa un desempeño excelente con fondo de textura y errores pequeños. Sin embargo, la luminosidad afecta a la propuesta, principalmente al componente de estimación de profundidad por lo que se

ha visto un error de 0.4 FPR. Por lo que, se puede concluir que sea realizado buenos algoritmos de percepción del objetivo en el plano de la cámara.

Al analizar todas las pruebas obtenidas, podemos concluir que el objetivo principal se ha cumplido, como está descrito en la propuesta que al tener una percepción bidimensional se puede realizar un movimiento tridimensional. Ya que, nosotros dividimos el problema en dos sub-problemas para realizar el movimiento en los dos planos: el plano de la cámara, que es la percepción bidimensional y el plano de la profundidad, estimándolo.

6.2 Problemas encontrados

En esta investigación se han detectado varios errores que se han mejorado de forma continua en todo el desarrollo de la propuesta correspondiente. El principal problema fue el modelamiento de Denavit-Hartenberg, esto se debe que el brazo manipulador estaba desalineado en los ejes de movimiento en las últimas articulaciones, por lo que se tuvo que rehacer el modelado del brazo manipulador. Además, por las desviaciones que no se han podido medir con exactitud ha afectado los resultados finales, pero con errores inferiores que no han afectado mucho al sistema de control.

Luego, al hacer uso de *Robot Operating System* se mostraron errores en la comunicación de los motores del brazo manipulador con la librería *default* de este *framework*. Por lo que, se investigó arduamente en solucionar el problema, que al final se resolvió cambiando el protocolo de comunicación del archivo de configuración del *framework*.

Adicionalmente, se tuvo problemas con los motores del brazo manipulador, ya que se realizó una sobrecarga a los motores, lo que averió los motores del *shoulder*. Por lo que, se tuvo que realizar pruebas con un brazo manipulador en un simulador llamado *framework Gazebo*.

Por último, otro problema que se ha presentado es en la percepción del objeto con iluminación cambiante. Por lo que, este brazo manipulador sólo puede estar en un ambiente estructurado con iluminación constante, es decir, iluminación ya sea clara o oscura.

6.3 Recomendaciones

Se recomienda investigar la parte teórica de brazos manipuladores, ya que para realizar el sistema de control de un brazo robótico se debe determinar el modelado cinemático y notación que se va a emplear. Además, al observar las diferentes técnicas del sistema de control con retroalimentación sensorial se debe tomar la idea de los brazos manipuladores que se utilizarán para realizar una investigación al respecto.

Luego, es útil realizar pruebas de los diferentes módulos para corregir los errores posibles y verificar que cada componente sea correcto. Por experiencia, es lo más adecuado para no volver a regresar al comienzo buscando los posibles errores que se han cometido.

Por último, se recomienda observar el número y el tipo de cada una de las articulaciones

del brazo manipulador, ya que dependiendo de estos criterios se escoge la cinemática inversa. Además, de la parte mecánica hay que cuidar la sobrealimentación de los motores, con un controlador de energía.

6.4 Limitaciones

Este trabajo presenta limitaciones en la parte de percepción, ya que tenemos las siguientes condiciones:

- Ambiente controlado.
- Objetivos específicos.

Por lo que, se tendría que modificar la parte de percepción de la propuesta, ya que trabajamos con diferentes técnicas que sólo funcionan sobre las condiciones presentadas.

6.5 Trabajos futuros

Se observa varias vertientes que se pueden trabajar, la primera vertiente es el sistema de visión en el cual se puede desarrollar diferentes técnicas para detectar diferentes objetos y realizar pruebas de seguimiento de trayectoria, además se puede observar el comportamiento de cada técnica en diferentes tipos de iluminación.

La segunda vertiente, son los modelos del sistema de visión, es decir, realizar comparaciones de las técnicas monoculares, binoculares y trioculares. Se puede analizar el costo computacional de cada modelo, además de analizar las ventajas y desventajas de cada modelo.

Otra vertiente, se relaciona al sistema de visión monocular para realizar la estimación de profundidad. Es decir, mejorar el sistema para tener un alcance parecido al de haber utilizado un sensor de profundidad. Adicionalmente, comparar el costo computacional que ambos sensores tienen para detectar un objeto.

Realizar una investigación profunda en las diferentes técnicas de cinemática inversa, especialmente en un sistema de control dinámico. Por lo que, se puede realizar pruebas y comparaciones de cada una de las técnicas, para luego detallar las ventajas y desventajas.

Por último, se puede ampliar la propuesta para un brazo manipulador a dos brazos robóticos en paralelo, para obtener una mejor estimación de profundidad por la sobre-posición de las imágenes tomadas por los brazos, de esta forma disminuir el error de estimación.

Bibliografía

- [Abadianzadeh et al., 2016] Abadianzadeh, F., Derhami, V., and Rezaeian, M. (2016). Visual servoing control of robot manipulator in 3d space using fuzzy hybrid controller. In *Robotics and Mechatronics (ICROM), 2016 4th International Conference on*, pages 61–65. IEEE.
- [Ajwad et al., 2014] Ajwad, S. A., Ullah, M. I., Islam, R. U., and Iqbal, J. (2014). Modeling robotic arms - a review and derivation of screw theory based kinematics. *Int. Conf. Engineering and Emerging Technologies (ICEET)*.
- [Alabdo et al., 2016] Alabdo, A., Pérez, J., Garcia, G. J., Pomares, J., and Torres, F. (2016). Fpga-based architecture for direct visual control robotic systems. *Mechatronics*, 39:204–216.
- [Alepuz et al., 2016] Alepuz, J. P., Emami, M. R., and Pomares, J. (2016). Direct image-based visual servoing of free-floating space manipulators. *Aerospace Science and Technology*, 55:1–9.
- [Ali T. Hasan and Isa, 2011] Ali T. Hasan, H. M. A.-A. and Isa, A. A. M. (2011). Neural network’s based inverse kinematics solution for serial robot manipulators passing through singularities. In Suzuki, K., editor, *Artificial Neural Networks - Industrial and Control Engineering Applications*, chapter 22, pages 459–478. InTech.
- [Alsina, 2014] Alsina, J. (2014). Introducción a la robótica. Producción Académica de Maestría.
- [Amouri et al., 2017] Amouri, A., Mahfoudi, C., Zaatri, A., Lakhal, O., and Merzouki, R. (2017). A metaheuristic approach to solve inverse kinematics of continuum manipulators. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 231(5):380–394.
- [Azizian et al., 2014] Azizian, M., Najmaei, N., Khoshnam, M., and Patel, R. (2014). Visual servoing in medical robotics: a survey. part i: endoscopic and direct vision imaging – techniques and applications. *The Intenational Journal of Medical Robotics and Computer Assisted Surgery*, 10(3):263–274.
- [Azizian et al., 2015] Azizian, M., Najmaei, N., Khoshnam, M., and Patel, R. (2015). Visual servoing in medical robotics: a survey. part ii: tomographic imaging modalities – techniques and applications. *The Intenational Journal of Medical Robotics and Computer Assisted Surgery*, 11(1):67–79.
- [Bagnell et al., 2012] Bagnell, J. A., Cavalcanti, F., Cui, L., Galluzzo, T., Hebert, M., Kazemi, M., Klingensmith, M., Libby, J., Liu, T. Y., Pollard, N. S., Pivtoraiko, M., Valois, J., and Zhu,

- R. (2012). An integrated system for autonomous robotics manipulation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 2955–2962.
- [Ball, 1900] Ball, S. R. S. (1900). *A treatise on the theory of screws*. Cambridge: University Press.
- [Barrientos et al., 1997a] Barrientos, A., Peñín, L. F., Balaguer, C., and Aracil, R. (1997a). *Fundamentos de Robótica*, chapter Morfología del Robot. McGraw-Hill.
- [Barrientos et al., 1997b] Barrientos, A., Peñín, L. F., Balaguer, C., and Aracil, R. (1997b). *Fundamentos de Robótica*, chapter Cinemática del Robot. McGraw-Hill.
- [Belfiore and Benedetto, 2016] Belfiore, N. P. and Benedetto, A. D. (2016). Connectivity and redundancy in spatial robots. *The International Journal of Robotics Research*, 19(12):1245–1261.
- [Broida and Chellappa, 1991] Broida, T. and Chellappa, R. (1991). Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 13(6):497–513.
- [Bruyninckx and Schutter, 2001] Bruyninckx, H. and Schutter, J. D. (2001). Introduction to intelligent robotics. unpublished notes for the “Robotics” course.
- [Bueno-López and Arteaga-Pérez, 2011] Bueno-López, M. and Arteaga-Pérez, M. A. (2011). Fuzzy logic control of a robot manipulator in 3d based on visual servoing. *IFAC Proceedings Volumes*, 44(1):14578–14583.
- [Cai et al., 2016] Cai, C., Somani, N., and Knoll, A. (2016). Orthogonal image features for visual servoing of a 6-dof manipulator with uncalibrated stereo cameras. *IEEE Transactions on Robotics*, 32(2):452–461.
- [Cañas et al., 2014] Cañas, J. M., Martín, L., and Vega, J. (2014). Innovating in robotics education with gazebo simulator and jderobot framework. *CUIEET 2014, XXII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*.
- [Chen et al., 2015] Chen, Y.-T., Shih, C.-L., and Chen, G.-T. (2015). An fpga implementation of a robot control system with an integrated 3d vision system. *Smart Science*, 3(2):100–107.
- [Corke, 2017] Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, volume 118. Springer.
- [Corke, 1994] Corke, P. I. (1994). Visual control of robot manipulators – a review. In *Visual Servoing*, pages 1–31. World Scientific.
- [Corke, 1997] Corke, P. I. (1997). *Visual Control of Robots: High-Performance Visual Servoing*. John Wiley & Sons, Inc., New York, NY, USA.
- [Corke and Hutchinson, 2001] Corke, P. I. and Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE T. Robotics and Automation*, 17(4):507–515.
- [Craig, 2005] Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Prentice Hall, 3th edition.

- [Craig, 2006] Craig, J. J. (2006). *Robótica*. Pearson Educación, 3th edition.
- [Ding et al., 2014] Ding, W.-H., Deng, H., Li, Q.-M., and Xia, Y.-M. (2014). Control-orientated dynamic modeling of forging manipulators with multi-closed kinematic chains. *Robotics and Computer-Integrated Manufacturing*, 30(5):421–431.
- [Dosovitskiy et al., 2015] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766.
- [Duka, 2014] Duka, A.-V. (2014). Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Procedia Technology*, 12:20–27.
- [Fischler and Firschein, 1987] Fischler, M. A. and Firschein, O., editors (1987). *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [Goertz, 1964] Goertz, R. C. (1964). Manipulator systems development at anl. *Proc. 12th Conf. on Remote Systems Technology, American Nuclear Society*, 12.
- [Hanmandlu et al., 1997] Hanmandlu, M., Shantaram, V., and Sudheer, K. (1997). Depth estimation from a sequence of images using spherical projection. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, USA, April 20-25, 1997*, pages 2264–2269.
- [Hasan and Abdellatif, 2012] Hasan, M. and Abdellatif, M. (2012). Monocular depth from motion using a new closed-form solution. In *Intelligent Robotics and Applications - 5th International Conference, ICIRA 2012, Montreal, QC, Canada, October 3-5, 2012, Proceedings, Part III*, pages 473–483.
- [Horn and Schunck, 1981] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1):185–203.
- [House and Road, 2014] House, F. and Road, I. (2014). Navy research and rescue. A Hydro-Lek Case Study.
- [Hutchinson et al., 1996] Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670.
- [Iliukhin et al., 2017] Iliukhin, V., Mitkovskii, K., Bizyanova, D., and Akopyan, A. (2017). The modeling of inverse kinematics for 5 dof manipulator. *Procedia Engineering*, 176:498–505.
- [Issac et al., 2016] Issac, J., Wüthrich, M., Cifuentes, C. G., Bohg, J., Trimpe, S., and Schaal, S. (2016). Depth-based object tracking using a robust gaussian filter. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 608–615. IEEE.
- [Jang et al., 1991] Jang, W., Kim, K., Chung, M. J., and Bien, Z. (1991). Concepts of augmented image space and transformed feature space for efficient visual servoing of an "eye-in-hand robot". *Robotica*, 9:203–212.

-
- [Jose et al., 2014] Jose, J. A. C., Cabatuan, M. K., Dadios, E. P., and Lim, L. A. G. (2014). Depth estimation in monocular breast self-examination image sequence using optical flow. *Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014 International Conference on*, pages 1–6.
- [Kamaliya and Patel, 2016] Kamaliya, P. K. and Patel, Y. K. D. (2016). Kinematic modeling and function generation for non-linear curves using 5r double arm parallel manipulator. *Journal of The Institution of Engineers (India): Series C*, 97(1):33–40.
- [Kazemi et al., 2012] Kazemi, M., Valois, J.-S., Bagnell, J. A., and Pollard, N. S. (2012). Robust object grasping using force compliant motion primitives. In *Robotics: Science and Systems*.
- [Kemp et al., 2007] Kemp, C. C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):20–29.
- [Keshavan and Humbert, 2017] Keshavan, J. and Humbert, J. S. (2017). An optical flow-based solution to the problem of range identification in perspective vision systems. *Journal of Intelligent & Robotic Systems*, 85(3-4):651–662.
- [KöKer, 2013] KöKer, R. (2013). A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. *Information Sciences*, 222:528–543.
- [Košecká et al., 1995] Košecká, J., Christensen, H. I., and Bajcsy, R. (1995). Discrete event modeling of visually guided behaviors. *International Journal of Computer Vision*, 14(2):179–191.
- [Kragic and Christensen, 2002] Kragic, D. and Christensen, H. I. (2002). Survey on visual servoing for manipulation. Technical Report ISRN KTH/NA/P-02/01-S, Centre for Autonomous Systems, Numerical Analysis and Computer Science.
- [Kucuk and Bingul, 2014] Kucuk, S. and Bingul, Z. (2014). Inverse kinematics solutions for industrial robot manipulators with offset wrists. *Applied Mathematical Modelling*, 38(7):1983–1999.
- [Leite and Lizarralde, 2016] Leite, A. C. and Lizarralde, F. (2016). Passivity-based adaptive 3d visual servoing without depth and image velocity measurements for uncertain robot manipulators. *International Journal of Adaptive Control and Signal Processing*, 30(8-10):1269–1297.
- [Lin et al., 2016] Lin, C.-Y., Hsieh, P.-J., and Chang, F.-A. (2016). Dsp based uncalibrated visual servoing for a 3-dof robot manipulator. In *Industrial Technology (ICIT), 2016 IEEE International Conference on*, pages 1618–1621. IEEE.
- [Liu et al., 2017] Liu, F., Gao, G., Shi, L., and Lv, Y. (2017). Kinematic analysis and simulation of a 3-dof robotic manipulator. In *Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference on*, pages 1–5. IEEE.
- [Liu et al., 2014] Liu, F., Shen, C., and Lin, G. (2014). Deep convolutional neural fields for depth estimation from a single image. *CoRR*.

- [Liu et al., 2016] Liu, F., Shen, C., Lin, G., and Reid, I. (2016). Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039.
- [Lovon-Ramos et al., 2016] Lovon-Ramos, P. W., Ripas-Mamani, R., Rosas-Cuevas, Y., Tejada-Begazo, M., Mogrovejo, R. M., and Barrios-Aranibar, D. (2016). Mixed reality applied to the teleoperation of a 7-dof manipulator in rescue missions. In *Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016 XIII Latin American*, pages 299–304. IEEE.
- [Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application in stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’81*, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Malis, 2002] Malis, E. (2002). Survey of vision-based robot control. In *ENSIETA European Naval Ship Design Short Course*.
- [Mancini et al., 2016] Mancini, M., Costante, G., Valigi, P., and Ciarfuglia, T. A. (2016). Fast robust monocular depth estimation for obstacle detection with fully convolutional networks. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4296–4303. IEEE.
- [Matthies et al., 1993] Matthies, L., Szeliski, R., and Kanade, T. (1993). *Kalman Filter-based Algorithms for Estimating Depth from Image Sequences*, pages 87–130. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Mehta, 2007] Mehta, S. S. (2007). Vision-based control for autonomous robotic citrus harvesting. Master’s thesis, University of Florida.
- [Nadi et al., 2014] Nadi, F., Derhami, V., and Rezaeian, M. (2014). Visual servoing control of robot manipulator with jacobian matrix estimation. In *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pages 405–409.
- [O’Kane, 2013] O’Kane, J. M. (2013). *A Gentle Introduction to ROS*. Independently published.
- [Osa et al., 2010] Osa, T., Staub, C., and Knoll, A. (2010). Framework of automatic robot surgery system using visual servoing. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan*, pages 1837–1842.
- [Palou and Salembier, 2012] Palou, G. and Salembier, P. (2012). *Computer Vision – ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part III*, chapter 2.1 Depth Estimation Of Frames In Image Sequences Using Motion Occlusions, pages 516–525. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Paul, 1984] Paul, R. P. (1984). *Robot Manipulators: Mathematics, Programming, and Control*, chapter Introduction. MIT Press, Cambridge, MA, USA, 6th edition.
- [Peters, 2017] Peters, J. F. (2017). *Foundations of Computer Vision: Computational Geometry, Visual Image Structures and Object Shape Detection*, volume 124. Springer.

- [Pinto et al., 2017] Pinto, A. M., Costa, P. G., Correia, M. V., Matos, A. C., and Moreira, A. P. (2017). Visual motion perception for mobile robots through dense optical flow fields. *Robotics and Autonomous Systems*, 87:1–14.
- [Qi et al., 2014] Qi, P., Liu, H., Seneviratne, L., and Althoefer, K. (2014). Towards kinematic modeling of a multi-dof tendon driven robotic catheter. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 3009–3012. IEEE.
- [Quigley et al., 2015] Quigley, M., Gerkey, B., and Smart, W. D. (2015). *Programming Robots with ROS*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- [Rahmatillah et al., 2013] Rahmatillah, A., Trilaksono, B. R., and Hindersah, H. (2013). 2.5-d visual servoing experiment using adept viper s850. *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, 10.
- [Ramirez, 2011] Ramirez, A. P. C. (2011). Control no lineal de un robot esférico de 3gdl con técnicas de control de linealización exacta. Informe Final de Trabajo no Publicado.
- [Raudies, 2013] Raudies, F. (2013). Optic flow. *Scholarpedia*, 8(7):30724.
- [Revaud et al., 2015] Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2015). Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172.
- [RoboTiCan, 2013] RoboTiCan (2013). *Komodo Robot Manual*. RoboTiCan, 1 edition. Innovative Robotics Engineering Solutions.
- [Shih et al., 2017] Shih, C.-L., Hsu, J.-H., and Chang, C.-J. (2017). Visual feedback balance control of a robot manipulator and ball-beam system. *Journal of Computer and Communications*, 5(09):8.
- [Spong et al., 2004a] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2004a). *Robot Dynamics and Control*, chapter Forward Kinematics: The Denavit-Hartenberg Convention. John Wiley & Sons, Inc., second edition edition.
- [Spong et al., 2004b] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2004b). *Robot Dynamics and Control*, chapter Introduction. John Wiley & Sons, Inc., second edition edition.
- [Thomas et al., 2002] Thomas, U., Maciuszek, I., and Wahl, F. M. (2002). A unified notation for serial, parallel, and hybrid kinematic structures. In *ICRA*, pages 2868–2873. IEEE.
- [Thota et al., 2013] Thota, S. D., Vemulapalli, K. S., Chintalapati, K., and Gudipudi, P. S. S. (2013). Comparison between the optical flow computational techniques. *Intenational Journal of Engineering Trends and Techonology (IJETT)*, 4(10):4507–4511.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- [Veitschegger and Wu, 1987] Veitschegger, W. K. and Wu, C. (1987). A method for calibrating and compensating robot kinematic errors. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, USA, March 31 - April 3, 1987*, pages 39–44.

- [Vihonen et al., 2016] Vihonen, J., Honkakorpi, J., Tuominen, J., Mattila, J., and Visa, A. (2016). Linear accelerometers and rate gyros for rotary joint angle estimation of heavy-duty mobile manipulators using forward kinematic modeling. *IEEE/ASME Transactions on Mechatronics*, 21(3):1765–1774.
- [Wang et al., 2016] Wang, F., Song, L., and Liu, Z. (2016). Image-based visual servoing control for robot manipulator with actuator backlash. In *Informative and Cybernetics for Computational Social Systems (ICCSS), 2016 3rd International Conference on*, pages 272–276. IEEE.
- [Weinzaepfel et al., 2013] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392.
- [Yang et al., 2014] Yang, H., Jin, M., Xie, Z., Sun, K., and Liu, H. (2014). Ground verification of space robot capturing the free-floating target based on visual servoing control with time delay. *Industrial Robot: An International Journal*, 41(6):543–556.
- [Yen and Huang, 1983] Yen, B. and Huang, T. (1983). Determining 3-d motion and structure of a rigid body using straight line correspondences. In *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 118–121.
- [Yilmaz et al., 2006] Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.*, 38(4):1–45.
- [Zhang et al., 2017] Zhang, Z., Bieze, T. M., Dequidt, J., Kruszewski, A., and Duriez, C. (2017). Visual servoing control of soft robots based on finite element model. In *IROS 2017-IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [Zhuang et al., 1994] Zhuang, H., Sudhakar, R., and Yu Shieh, J. (1994). Depth estimation from a sequence of monocular images with known camera motion. *Robotics and Autonomous Systems*, 13(2):87–95.